User's Manual
Huber Slit Controller
Model HSC-1

X-ray Instrumentation Associates

8450 Central Ave
Newark, CA 94560 USA

Tel: (510) 494-9020; Fax: (510) 494-9040
http://www.xia.com

Information furnished by X-ray Instrumentation Associates (XIA) is believed to be accurate and reliable. However, no responsibility is assumed by XIA for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of XIA. XIA reserves the right to change specifications at any time without notice

Copyright 1998 by X-ray Instrumentation Associates

Manual # HSC-MAN-001.4.1
January 27, 2004

1 Introduction

The HSC-1 stepper motor controller from X-ray Instrumentation Associates (XIA) is designed to provide a flexible and easy way to control Huber Slits. All motor control is performed locally by the small, self-contained HSC-1 unit that attaches directly to the Huber slit assembly. Computer control is very easy, using a simple set of ASCII commands sent through a standard RS-232 serial port; even a ‘dumb’ terminal can be used to communicate with the HSC-1 controller. The slits can also be controlled directly using buttons on the controller printed-circuit board. One serial port can be used to communicate with many HSC-1 devices; the upper limit depends on the specific installation. Note that the serial port does not provide the power to drive the devices; a power supply kit is available separately to provide power for up to four HSC-1 devices. This kit (model HSC-PWR) consists of a desktop power supply and a special adapter cable, which converts a standard 9-pin RS-232 connector to the special pin-out used by the HSC-1. Installations with more than 4 HSC-1 devices require multiple power supplies and adapters.

The language used to control the HSC-1 controllers is simple yet very flexible. The commands are easy to use and remember, making manual control using a terminal (or terminal window) a real option for a simple experimental setup.

1.1 Summary of Features

- One HSC-1 drives one axis of Huber slits.
- Low cost – no other control circuitry required
- Low noise, low power, 2.5 μm/step
- Onboard micro-controller and motor drivers
- Simple RS-232 command language
- Wall transformer powers up to 4 HSC units
- Power and RS-232 signals daisy-chained between units, reducing cabling requirements
- User controlled firmware motion limits
- Current location and parameters saved on power failure
- Pushbuttons allow manual motor control while letting the controller track motion
- Safe torque levels – no danger of damage to slits

1.2 Summary of Installation Procedure

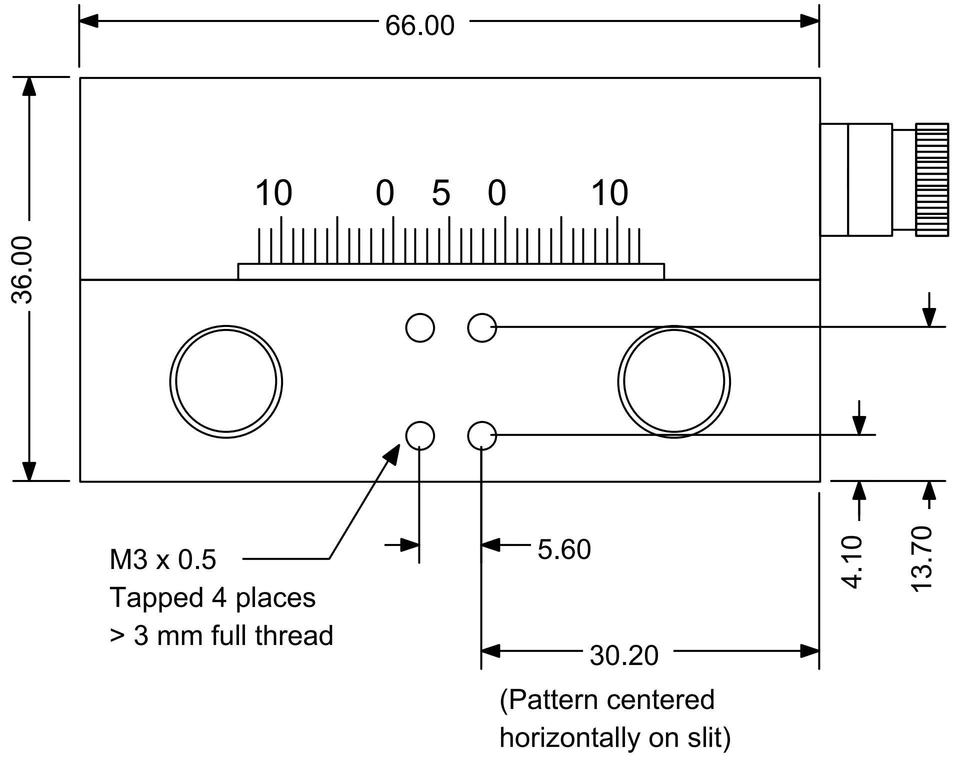
- Mechanical installation of the HSC-1 (see Section 2)
- Make signal and power electrical connections and set jumper (see Section 3)
- Check communication between host and each HSC-1. Assign aliases if desired.
- Ensure that the arbitration priority is different for each module (see section 5 for a description of arbitration priority, and section 6 for the method used to change its value).
- Go through the calibration procedure for each HSC-1 unit (see Section 4.3)
- After calibration, the controllers are ready to use!

2 Mechanical Installation

The HSC-1 provides a simple and inexpensive way to control Huber slit assemblies. Installation is fast and easy. If your Huber slits do not possess the necessary mounting holes, a diagram of the required hole pattern is shown in Figure 1. Also note that two knob styles are supported; recent Huber slits have knobs measuring 0.425 inches in diameter (measured at the knurl), while some older slits have knobs with a diameter of 0.390 inches. Measure the diameter of the couplers on the stepper motor shafts to determine whether they match the corresponding knobs; the coupler that fits the new larger knobs has an outside diameter of 0.600 inches, while the smaller coupler measures 0.563 inches in diameter. If you have the wrong type of couplers, please contact XIA to arrange for replacements.

Please note that if you purchased your HSC-1 controllers already mounted on Huber slits, you can skip the mechanical installation and proceed directly to making the electrical connections, described in Section 3.

Figure 1: HSC-1 standoff mounting hole pattern; all dimensions are in millimeters.



2.1 List of Parts in Standard unit, as shipped

1. HSC-1 assembly, containing 2 stepper motors and controlling circuitry
2. Black anodized aluminum mounting post
3. Aluminum jig to locate mounting post on the Huber slit assembly
4. Four M3x25 socket-head screws to attach the standoff to the Huber slit assembly
5. Two 6-32x3/8" bolts to attach the HSC assembly to the mounting post
6. Two helical shaft couplers to connect the motor shaft to the control knobs on the Huber slit assembly
7. 2.5 mm hex key

2.2 Mechanical Assembly Procedure

1. First, mount the standoff on the Huber slit assembly, using the 4 socket-head metric screws, which require a 2.5 mm Allen key (included). Locate the orientation dimple on the standoff adjacent to one of the holes used to mount the HSC-1 assembly. The standoff should be oriented so that the marked mounting hole is over the position scale of the slits, not hanging over the edge of the slit assembly. If the coupler is installed backwards, the motors will not be centered over the knobs, and binding may result. Use the Aluminum mounting jig to center the mounting post between the knobs (see separate instructions).
2. Place the helical couplers over the knobs, and tighten the set-screws with a .050" hex key (not included yet). Make sure that the coupler is sitting flat on the knob, and try to tighten the set-screws evenly, so that the coupler is centered over the knob. The coupler should fit fairly snugly over the knob; if there is a lot of slop or if the coupler does not fit over the knob, you probably have the wrong coupler (there are two different knob sizes). Contact XIA to exchange the coupler for the correct one.
3. Now place the HSC controller onto the mounting post, sliding the motor shafts into the couplers and the dowels in the bottom of the mounting plate into their locating holes in the mounting post. Everything should line up; if the alignment is not good, check the position of the mounting post as well as the position of the couplers on the knobs. The assembly will fit in either direction (connectors in or out). Note that Motor A is on the left as you look at the front of the case (the side with connectors).

4. Thread the 2 6-32x3/8” bolts up through the mounting post into the motor mounting plate to secure the HSC controller to the post. Tighten the bolts with a ¼” box-end wrench (not included yet).
5. Tighten the couplers onto the motor shafts with a 3/32” hex key (also not included yet).
6. Rotate the couplers and make sure there is no binding and that the mounting post does not interfere with the set-screws. If the set-screws rub against the mounting post, be sure the set-screws are tight and that the mounting post is well centered between the knobs.

3 Electrical Connections

The electrical connections to the HSC-1 are not standard RS-232, since power must be provided along with the serial data. The pin-out of the HSC-1 connectors is summarized below. This detailed information is important only for a custom installation or for debugging purposes; if the HSC-PWR power adapter (described below) is used, connecting the HSC-1 controller to a standard serial port is as easy as plugging in a cable.

Pin Number	Connection
1	Power
2	RxD (Serial data to Host)
3	TxD (Serial data from Host)
4	Power
5	Ground
6	Power
7	Ground
8	Ground
9	RxD Arbitration

Table 1 : HSC-1 pin assignments.

3.1 Signal Connections

The HSC-1 controller supports a standard 3-wire DTE (Data Terminal Equipment) – DCE (Data Communications Equipment) connection, with no support for hardware flow control or modem control. Pin 2 (RxD) carries serial data from the HSC-1 devices to the host, while Pin 3 (TxD) carries serial data from the host to the daisy-chained devices. A signal ground connection (Pin 5 on a standard 9-pin serial port connector) completes the 3-wire connection. All connections are straight through; no twist (as found in a null-modem DTE-DTE connection) is needed between Pins 2 and 3. All devices simultaneously monitor the TxD line for commands from the host, while use of the RxD line is shared between all devices. Pin 9 carries an arbitration signal, used to prevent garbled responses which might arise from multiple devices attempting to transmit simultaneously. The HSC-1 is set to run at 9600 baud, with 8 data bits, 1 stop bit and no parity.

If the HSC-PWR adapter module is installed between the host and the first HSC-1 device, Pins 7 (RTS – Request To Send) and 8 (CTS -- Clear To Send) are connected together on the host side to provide a hardware flow control loop-back. In addition, Pins 1 (DCD – Data Carrier Detect), 4 (Data Terminal Ready) and 6 (DSR – Data Set Ready) are tied together on the host side to form a modem control loop-back. This helps to ensure that the serial connection will work for all flow-control settings of the serial port in use. When the adapter module is installed between daisy-chained HSC devices, Pins 7 and 8 tie together Ground while Pins 1, 4 and 6 tie together the three Power connections. The Ground connection to Pin 5 is used to connect the signal ground between the two devices.

To connect devices where there is no need for a power adapter, just use a 9-conductor straight through cable with female DB9 ends; note that all 9 conductors must be present. Special cable is not necessary – easy-to-use flat ribbon cable is recommended.

3.2 Power Connections

Six conductors are used to carry power to the HSC devices, three each for power and ground; see the Table above. Only one supply voltage is needed; on each board, this voltage is filtered and used to drive the motors, as well as to produce the 5 Volts used by the digital logic. The filtering keeps the noise from the motor drivers' switching power supplies off of the common supply bus. The power requirements for a single HSC-1 module are listed below:

- Supply Voltage Range: 24 – 30 Volts. Voltages lower than 24 Volts may be used, with less immunity to accidentally triggering the power failure sensing circuit (which uses a threshold of 18 Volts)

Supply Current:	50 mA with no motors moving
	175 mA with one motor moving (motor current set to high)
	300 mA with both motors moving (motor current set to high)

The easiest way to connect power to an HSC system is to use the HSC-PWR adapter cable and associated power supply; one supply and adapter module is needed for every four HSC-1 modules. One end of the 6 foot adapter cable (female DB9, labeled HSC-PWR) plugs directly into the first HSC-1 controller in a daisy chain; thumbscrews are provided to secure the connector to the HSC-1. The other end of the adapter cable incorporates both a male DB9 connector (with jackscrews) for making the connection to the host serial port, and a power jack. To provide power to the HSC-1, just insert the round plug at the end of the cable from the power supply module into the power jack.

The power adapter can also be used to split a large daisy chain into small enough segments to be powered by a single supply. In this case, a power adapter cable should be plugged into the input connector of the first device in each segment, and the associated power supply plugged into the power jack at the other end of the cable. A standard 9-conductor cable should be run between the output connector of the last device in the upstream segment to the input side of the power adapter. The power adapter breaks the power connection between segments (allowing the use of different supplies) while keeping the data and arbitration line connections continuous across the whole daisy chain.

Whether or not the HSC-PWR supplies are used, care must be taken not to connect too many devices to a power daisy-chain. It is important not to surpass the current-carrying capacity of the cables connecting the devices, or the printed-circuit board trace carrying the power between the input and output connectors. Under no circumstances should a power daisy-chain include more than eight HSC devices; if a single supply is used to power more than eight devices, split the power daisy-chain into several groups of less than eight devices, and make a separate power connection to each group of devices.

3.3 Arbitration

As specified above, pin 9 carries an arbitration signal, used to prevent garbled responses due to simultaneous transmission by multiple HSC devices. This line functions as a wire-or between all devices; any HSC device can pull the line low in order to reserve the line. The arbitration line must be connected to all HSC devices, but isolated from the host. This is controlled by the setting of jumper JP1 located in the corner of the PC board adjacent to the input DB9 connector. The jumper has two possible positions (there are three pins) – the "On" position corresponds to connecting the two pins furthest from the edge of the board. In the "on" position, the Arbitration line is connected to pin 9 of the input connector. In the "Off" position (the two pins closest to the edge of the board are connected), pin 9 of the input connector is isolated from the arbitration line. For proper operation, the jumper positions on the HSC devices should be set as follows:

- On the first HSC device in the chain (closest to the host), set JP1 to the "Off" position.
- On all other devices, keep JP1 in the "On" position.

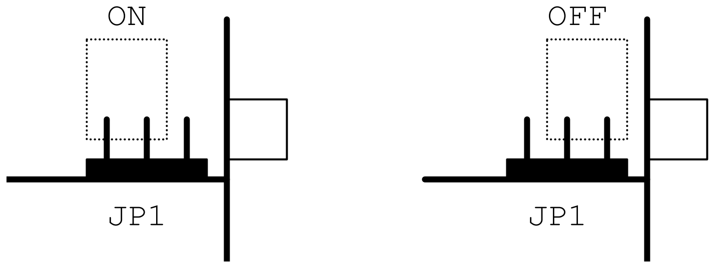


Figure 2: Jumper JP1, looking from the input connector side of the HSC.

There is a small amount of time between when a device tests the arbitration line to see if the RxD line is available and when the line is pulled low; this allows situations where multiple devices can pull the arbitration line low. To avoid garbled responses from such situations, a second level of arbitration is used: after an amount of time specific to a particular device, the arbitration line is released momentarily. If the line is pulled high by the pull-up resistor, then the line is available, and the device pulls the line down again and starts transmitting. If the line stays low, another device is holding the line down; in this case, the device releases the line and waits for the line to become available (as indicated by the arbitration line going high again). The pull-up resistor is fairly large (27 kΩ - one per board), in order to conserve power and allow a large number of devices on the chain. For systems involving a small number of HSC devices and long cables, the rise-time of the arbitration signal can be quite long (tens of microseconds). The maximum rise-time allowed is 30 µsec, which roughly corresponds to a 50-foot (15-meter) length of ribbon cable per device. If the rise-time of the arbitration line is longer than that, then no device will be able to grab the RxD line, since it will always seem like another device is holding the arbitration line low. In that unlikely case, it will be necessary to install an additional pull-up resistor between the arbitration line and Vcc (+5 Volts); please contact XIA for assistance.

4 HSC Units, Coordinate System and Calibration

This section describes how the position of the Huber slits is represented in the HSC-1 controller.

4.1 Units

The HSC-1 measures all positions in terms of stepper motor steps (this applies to both the internally stored positions and the positions specified by the user). There are 200 steps per revolution for the stepper motors used in the HSC-1. One full rotation of a knob on the Huber Slit assembly moves the corresponding shutter 0.5 mm, so 1 mm of slit travel corresponds to 400 steps. To make the conversion clear and easy to find:

$$\boxed{1\,mm = 400\,steps}$$

4.2 Coordinate System and Limits

In the HSC-1, all positions are represented as a positive number of steps. Two bytes are used to store each position, so the possible range is 0 through 65535 steps. This corresponds to a maximum range of motion of greater than 160 mm, which is much larger than the full range allowed by the Huber slits; clearly, there must be some way to limit the motion to a smaller range.

The lower range of motion is always at 0 steps; this corresponds to the most inward allowable position of the slits. The upper range of motion is a variable parameter stored in HSC-1 memory; by default, the upper limit is set to 4400 steps, corresponding to 11 mm of total travel (see Section 6.3.13 for instructions on how to change this limit). With limits enabled, the HSC-1 will not move the slits beyond these limits (there is one exception to this rule – the single-step command, detailed in the following section concerning calibration). If limits are disabled, the HSC-1 will (try to) move the slits to any position specified by the user; care should be taken not to run the slits into each other or run them off the end of their screws.

The motor position (in steps) corresponding to the zero position of the slits is variable and can be located anywhere within the allowed range of motion (see Section 6.3.13). By default, the slit origin corresponds to 400 steps, which allows 1 mm of inward motion beyond the origin. Note that if a slit is moved more than 0.5 mm inward past the origin, it begins to leak *behind* the slit, *i.e.* there is a gap between the back of the slit and the frame of the slit assembly.

The following figure illustrates the coordinate system used by the HSC-1:

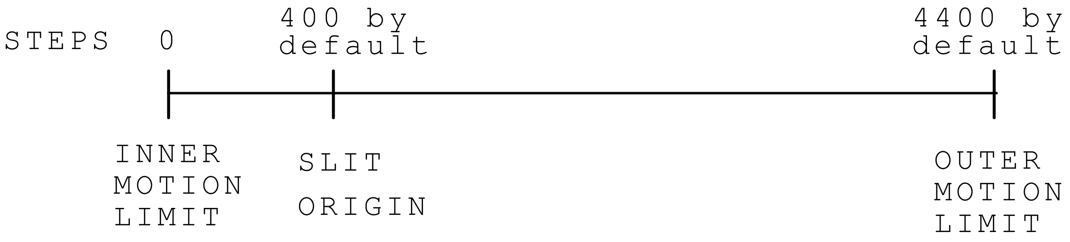


Figure 3: HSC coordinate system

4.3 Calibration

One of the most important aspects to understand about using the HSC to control Huber Slits is the fact that there is no position feedback from the slits to the motor controller. As a result, it is absolutely essential that the controller keep track of the slit locations at all times, so that the slits can be moved without running into each other or running them off their screws.

After mechanical and electrical installation, the HSC-1 controller does not know the position of the slits, and no movement is allowed. The HSC must be calibrated, which simply involves locating the origin (both slits set to 0) of the slits for the controller. There are two calibration modes available with the HSC-1: *manual* (where the user needs access to the HSC to press the buttons) and *immediate* (where the controller just sets the motors to their zero positions and sets the calibrated flag to true). Each mode will be discussed separately. See Section 6.3.1 for instructions on how to send the appropriate calibration commands. Note that both calibration modes set the motor positions to the values corresponding to the slit origins; there is no way to set the motor positions to an arbitrary value (which can be dangerous, if the motor positions are set to the wrong values). As long as limits are enabled, the controller will never move the slits closer together than their calibrated origins, so there is no danger of running the slits into each other with the calibration modes that are supported.

4.3.1 Manual Calibration

Manual calibration is the best way to make the stored motor zero positions correspond exactly to the origin of the slits. The procedure is described below; note that the calibration sequence is aborted if there is ever a thirty-second period in which no buttons are pressed. If the calibration sequence is aborted, the controller is left in an uncalibrated state, even if the controller was calibrated before initiating the calibration sequence.

- Send the command to initiate the manual calibration (see Section 6.3.1) to the HSC; all four LED's will flash
- Manually (by rotating the knobs/couplers), position each slit at the desired 'zero' position. Typically, this will be with the position of both slits set to zero (both on the indicating window and on the knob) and the slits touching. Under certain circumstances, however, a different 'zero' position may be desired. It may be easier to do this before sending the calibrate command, to avoid having the calibration sequence time-out if this process takes more than 30 seconds. Note that the 'zero' position defines the closest approach of the slits, since the controller assumes that they are touching when calibrated.
- Once the slits are both at zero, press any of the four buttons, located at the ends of the PC board, two on each side (labeled A-CW, A-CCW, B-CW and B-CCW). This step locates the origin for the motor controller; the following steps are used to make sure the motors are in phase with the controller. The step sequence runs through four phases; when the knob is manually set to the zero point, there is no guarantee that the motor phase will match the current phase stored in the controller. Without the following steps, the slit position could end up off by several steps.
- Once the button is pressed, the controller moves each slit out 0.5 mm (one turn), and then moves motor A back in one turn, nominally back to zero. While in motion, the LED's corresponding to the movement direction will be lit; when the motion is complete, the motor A LED's will be lit and the motor B LED's will be flashing.

  Note: if this is the initial calibration after installation, pay attention to the movement of the motor. The motor should move smoothly, with no skipping. Irregularity in the motor motion indicates either binding in the motor-knob coupling due to improper alignment or that the knob is hard to turn and is in need of lubrication. Check the alignment; if the alignment is good, a slower motor speed (see Section 6.3.13 and Table 2 for details on changing the motor speed) should be selected to get more torque to turn the sticky knobs. Service on the slits themselves may be necessary to ensure smooth movement, especially for older slits.

- Use the motor A buttons to reposition slit A to zero; just a few steps should be necessary (pressing and releasing a button will move the motor one step; if the button is held down for more than half a second, the motor will step repeatedly.) After slit A has been properly positioned, press one of the motor B buttons (next to the flashing LED's). Motor A will move out one turn, and motor B will move in one turn, bringing slit B close to the origin. Now the motor B LED's will be lit, and the motor A LED's will flash.
- Use the motor B buttons to position slit B to zero, then press one of the motor A buttons. Motor A will move in one turn to the origin, all four LED's will turn on briefly, and the calibration sequence will end. At this point, the motor positions in the HSC controller will be set to the value corresponding to the origin (nominally 400 steps), the calibrated flag will be set to true, and the completion response will be sent to the host.

4.3.2 Immediate Calibration

Immediate calibration is useful in circumstances where the buttons are not accessible, or if calibration with the beam on is desired. Immediate calibration requires no manual user input; immediately upon receiving the immediate calibration command, the motor positions are set to the value corresponding to the slit origin, and the calibrated flag is set to true.

To calibrate with the beam on, it may be necessary to move the motors before sending the command to perform the immediate calibration. For example, it may be useful to use the beam itself to indicate whether the slits are open, and to define the origin as the position where no beam passes through. If the HSC is not calibrated, the normal move commands are disabled; even if the HSC is calibrated, the slits cannot get closer together than the origin set in the previous calibration. For this reason, the HSC supports a single-step command, which allows the user to move either slit one step in either direction, even if the controller is not calibrated. Since this command does not check on the motion limits, care should be taken not to harm the slits.

4.3.3 Uncalibrate

Since the HSC-1 controller saves its calibration status when power is removed, it is necessary to provide a command to put the module into an uncalibrated state. This is useful in preventing problems if one of the knobs is turned manually which voids the calibration. It also provides a means of locking out any motion commands. Finally, if an HSC-1 unit is dismounted from a set of Huber slits, it should be put into an uncalibrated state, so that when it is reinstalled on another set of slits a new calibration sequence will be necessary before moving the slits.

5 HSC Operation and Setup

Many HSC operating parameters can be modified to fit a particular installation. The parameters are stored in internal memory in the HSC microcontroller, as well as in an external EEPROM so the setup is maintained even if power to the HSC is turned off. Parameters are available to control the motor operation, the motion limits and origin setting, and the serial communication with the HSC devices. A full memory map for the HSC is in Table 2 below, a detailed description of each parameter follows. See Section 6.3.13 for details on how to use the **W** (write) command to modify the parameters or the **R** (read) command to read their values.

- Outer Motion Limit: With limits enabled, this sets the maximum opening for each slit. By default, this limit is set to 4400 steps, which corresponds to 11 mm of slit travel.

- Origin Position: this is the value (in steps) that corresponds to the origin of the slits. By default, the origin position is set to 400 steps, which allows 1 mm of inward slit travel beyond the origin.

- Motor Position (A and B): these integer variables hold the internally stored motor positions, in steps. These memory locations are read-only; the only way the user can modify these positions is by performing a calibration.

- Motor Step Delay: this parameter controls the time taken for each step of the stepper motors. Each unit corresponds to a change in step time of about 40 μsec; the minimum step time is 1.2 ms (which corresponds to 833 steps/sec, or a slit speed of about 2 mm/sec). By default, the motor step delay is set to 100, for a step delay of 5.2 ms. Longer delay times are useful in providing the extra torque necessary to drive sticky knobs; if the knobs are well-lubricated and easy to turn, faster motor speeds are reliable. The maximum step delay setting is 255, which corresponds to a delay of 11.4 ms, or just 88 steps/sec (about ¼ mm/sec).

- Gear Backlash: This parameter sets the number of steps of backlash used by the HSC. To accurately position the slits, the HSC always finishes any move by moving the slits inward (motors move clockwise, step numbers decrease). The backlash parameter controls how far beyond the final position the motors are moved before reversing direction and moving to the desired position. By default, this value is set to 10 (1/20 turn).

Table 2: HSC-1 Memory Map

Index	Variable Name	Visibility	Range	Default Value
1	Outer Motion Limit	[RW]	[0 – 65535]	4400 (10 mm from default origin)
2	Origin Position	[RW]	[0 – 65535]	400
3	Motor A Position (steps)	[R]	[0 – 65535]	NA
4	Motor B Position (steps)	[R]	[0 – 65535]	NA
5	Motor Step Delay	[RW]	[0-255]	100 (roughly 5.2 ms/step)
6	Gear Backlash	[RW]	[0 – 255]	10
7	Control Word (see Table 3)	[RW]	[0 – 255]	142 (0b10001110)
8	Escape Character	[RW]	Non-character, printable ASCII	'!' (ASCII 33)
9	Arbitration Priority	[RW]	[0 – 255]	Set based on Serial number
10	Motor A Phase	[R]	Internal Use Only	
11	Motor B Phase	[R]	Internal Use Only	
12	Calibration Complete	[R]	Internal Use Only	
13	EEPROM Signature	[R]	Internal Use Only	
14	EEPROM Version	[R]	Internal Use Only	

- Control Word: This word contains several boolean switches, as well as a 2-bit field that controls the power level of the motors. The control word is described in Table 3 below. Note that the control word must be changed as a whole, by writing a decimal number into memory; the bits are not individually addressable. By default, the control word is set to 142, which corresponds to a high power level, limits enabled, print introductory banner true, and print error text true. By default, the buttons are not locked, the commands are not echoed, and the serial number is used for identification in the HSC responses.

- Escape Character: This is the character used to start any command from the host; until the HSC sees this character, all characters are ignored. The default escape character is '!', which corresponds to 33 ASCII. Any printable ASCII character can be used, except for characters ('a' – 'z', 'A' – 'Z'), numerals ('0' – '9'), '+' and '-', which are used in the body of commands. Any character used for the escape character should not be used as part of a module alias. Using the default escape character is strongly recommended.

- Arbitration priority: this value determines how long the device waits after initially grabbing the arbitration line before releasing it briefly to test if another device is also trying to talk and holding the line down. If the line is held down, the device testing the line releases the line and waits until it is available. If more than one device grab the line simultaneously, the device with the highest arbitration priority (the one that waits the longest before testing the line) will win control of the RxD line and talk first. Ideally, every device on the line will have a different arbitration priority. Each unit corresponds to a delay of 40 µsec, on top of a minimum delay of 40 µsec. By default, this parameter is set based on the serial number: a check-sum is formed from all characters in the serial number, then the lowest four bits are used for the arbitration priority, giving values from 0 through 15. If all devices on a daisy-chain have consecutive serial numbers, then the arbitration priorities should be unique (for fewer than 16 devices). The priorities can be changed manually if more than one device has the same priority, or to optimize the response time if desired.

Table 3: HSC-1 Control Word

Bit(s)	Variable Name	Range	Default	Description
0 – 1	Power Level	[0 – 2] (3 is mapped to 2)	2 (high power)	Controls motor current. 0 = low, 1 = medium, 2 = high.
2	Limits Enabled	0 = FALSE, 1 = TRUE	TRUE	Determines if motion limits are checked prior to moving motors.
3	Print Intro Banner	0 = FALSE, 1 = TRUE	TRUE	Determines whether the introductory banner is printed on startup. Can be turned off to avoid multiple copies from multiple devices.
4	Command Echo	0 = FALSE, 1 = TRUE	FALSE	Determines whether the commands from the host are echoed back. Should not be enabled in more than one device. Note: when echoing the command, the arbitration line is not checked.
5	Lock Buttons	0 = FALSE, 1 = TRUE	FALSE (Buttons enabled)	Determines whether the buttons on the HSC are enabled to move the motors.
6	Use Alias as Id	0 = FALSE, 1 = TRUE	FALSE	Determines whether the module alias or the module serial number is used for module identification in any responses.
7	Print Error Text	0 = FALSE, 1 = TRUE	TRUE	Determines whether a text description of an error is printed after the error code. Turning off the error text may be favored when under computer control.

All other memory variables are read-only, and used internally in the HSC.

5.1 Manual Motor Movement

The motors in the HSC can be moved manually, using the buttons on the ends of the controller PC board. Once the HSC is calibrated, **the knobs should never be used to move the slits**, since that destroys the calibration and could lead to damage to the slits. The buttons are only enabled when the HSC is calibrated and the Buttons Locked bit in the Control word is false. Any motion limits are in effect (if enabled).

If a button is pressed and released right away, the motor will move one step in the corresponding direction. Holding down the button for half a second will initiate repeat mode, where the motor moves continuously until the button is released. If a CCW button is pressed, backlash will be taken out half a second after the button is released. If the CCW button is pressed again sooner than a half-second after release, the backlash removal is delayed, so it is easy to position the motor without being confused by the backlash motion. To remove the backlash, the HSC just moves the motor the chosen number of steps CCW (slit moves outward), then immediately moves the motor the same number of steps CW (slits move inward). As stated above, the slit position is reliable as long as the position is always approached from the same direction.

5.2 Replacing an HSC-1 Module

Replacing a faulty HSC-1 module is not quite as easy as just installing a new one on the same pair of slits (or even installing a new pair of slits already instrumented with HSC-1 controllers). The main problem concerns addressing the replacement controller. If the controlling program uses the serial numbers to address the modules, then the program must be updated with the serial number of the replacement module (either by hand or by polling all of the devices using the **ALL** address and searching for new serial numbers). If the program addresses the modules using aliases, then the alias of the replacement module must be set to match the alias of the module being replaced. There are two basic approaches to setting the alias to match the desired alias:

- Set the new alias with the new module in its place on the daisy chain of devices. To set the alias in this situation, it is necessary for the program to know the serial number of the new module. This can be done either through user input, or by polling all the devices to get their serial numbers (using the **ALL** address and the Inquiry command, described below), and then comparing the returned serial numbers with the list of serial numbers previously in use. The serial number of the replacement module will not be known, and can be given the alias corresponding to the serial number that came up missing.

- The other approach to setting the alias avoids the problem of having to know the serial number. In this approach, the replacement module is installed temporarily as the only device on the serial bus; then the special **ALL** address described below can be used to set the alias. If the alias is used as the response Module-Id, then the control word (described above) should be set appropriately. The alias can even be set by hand using this approach, by using standard terminal software (Hyperterm, for example)on any PC to communicate to the replacement HSC-1 via the standard COM port.

6 HSC Command Language and Responses

This section describes the commands used to control the HSC, as well as the responses returned by the HSC to each command. The command syntax is easy to use, so that manual control through a terminal is a definite alternative to computer control.

In the following sections, command text is in boldface Arial font:

! A command to the HSC-1 looks like this

Note that command text is not case sensitive.

Response text from the HSC-1 controller is in Courier New font:

```
%A response from the HSC-1 looks like this;
```

6.1 Command and Response Syntax

The HSC uses a very simple command language, using single-character, easy to remember mnemonic commands. The basic command syntax is as follows:

!Module-Id Command Arg1 Arg2 <carriage return>

The Module-Id can be either the module serial number or the module alias. To address all modules in the daisy chain, use the special **ALL** Module-Id. Note that the Module-Id is not case sensitive.

The Command is a single character; see Table 4 below for a list of available commands, and Section 6.3 for a detailed description of each command. A space must separate the Command from the Module-Id.

The number of arguments depends on the command; the maximum number of arguments is two. No space is needed between the command and the first argument; a space must be used to separate numerical arguments. Single-character non-numerical arguments do not need to be separated by a space. See Table 4 and Section 6.3 for descriptions of the arguments for each command.

Table 4: HSC Command Summary

Command	Description	Argument 1	Argument 2
0 (zero)	Calibrate	Calibration type (M or I or -)	None
1 (one)	Single step	Motor (A or B)	Direction (+ or -)
A	Assign text alias	Alias (up to 24 characters)	None
C	Close slits	Change in opening (steps)	None
I	Inquire – returns module status	None	None
K	Kill movement	None	None
M: version 1.0 – 1.2	Absolute Move	New Motor A position (steps)	New Motor B position (steps)
M: version 1.3 or greater	Absolute or Relative Move	New motor position or number of steps to move	New motor position or number of steps to move
O	Open slits	Change in opening (steps)	None
P	Position inquiry	None	None
R	Read from internal memory	Index of variable	None
S	Slide slits	Direction (+ or -)	Change in center position (steps)
T	Test module	None	None
W	Write to internal memory	Index of variable	New value

The responses from the HSC take the following form:

```
%Module-Id Response text;<Carriage Return>
```

All responses will begin with the '%' character, followed immediately by the Module-Id. By default, the Module-Id is the module serial number. If an alias has been set (see command **A**), the alias can be chosen as the response Module-Id by setting the appropriate bit in the Control word. The response text is described below along with the detailed description of each command; note that there may be carriage returns in the response. The response is ended by a ';' character followed by a carriage return.

After receiving a command, the HSC-1 will immediately return a response indicating the status of the command. The three possible responses are `OK`, `BUSY` and `ERROR`. `OK` indicates that the command was received and is being processed. `BUSY` indicates that a motor movement is in process and the controller cannot process the command; the command must be resent after the motors complete their motion. Finally, `ERROR` (followed by an error code and optionally by a brief description of the error) indicates that there is a problem and the command cannot be processed.

6.2 Startup Responses

When power is first applied to the HSC, the module can produce several startup messages. If the module is not calibrated, the following message will be sent:

```
%Module-Id Uncalibrated!;
```

If the EEPROM data is not valid (due to an incomplete write to the EEPROM on the previous power-down cycle), the HSC will report

```
%Module-Id Invalid EEPROM! Loading defaults;
```

All memory variables will be set to their default values; any custom settings will have to be restored.

Finally, if Print Intro Banner is set to true in the Control word, the following response will be produced:

```
%Module-Id HSC v1.0 (c) XIA 1998 All Rights Reserved;
```

Note that the version number may differ from 1.0.

6.3 Detailed Language Summary

This section describes each command in detail, and lists the possible responses.

6.3.1 0 (zero): Calibrate

This command initiates a simple calibration sequence, which is used to tell the HSC-1 controller the position of the slits. This calibration is necessary prior to any motor movement, since there is no feedback of the slit position to the controller; the only way to ensure safe movement of the slits is to have the controller keep track of the position in internal memory. For this reason, the controller does not allow any motor movement (except for single steps – see command **1**) until the calibration sequence is successfully completed.

There are three calibration options: manual (M argument), immediate (I argument) and uncalibrate (-argument). The calibration procedures are described in detail in Section 4.3.

Manual Calibration:

Syntax: **!Module-Id 0 M**

Returns:	`%Module-Id [OK or BUSY];`	immediately
	`%Module-Id Apos Bpos DONE;`	after successful completion

Note: the returned positions Apos and Bpos will equal the Origin Position described above.

If the calibration sequence times out (more than 30 seconds elapse after a button is pressed without completing the calibration sequence), the following response is produced:

```
%Module-Id Timeout – CAL ABORTED!;
```

Example: **!Primary-Vertical-Slit 0 M**

```
%XIAHSC-B-0037 OK;
```

Calibration sequence is completed

```
%XIAHSC-B-0037 400 400 DONE;
```

```
Immediate Calibration:
```

```
Syntax:
```
!Module-Id 0 I

Returns:

```
%Module-Id Apos Bpos DONE;

where Apos = Bpos = slit origin position.
```

Uncalibrate:

Syntax: **!Module-Id 0 –**

Returns:

```
%Module-Id OK Uncalibrated;
```

6.3.2 1 (one): Single Step

This command moves either motor one step in either direction. This movement command is special: the motion limits are not checked, and the command works even if the unit is not calibrated. Care should be taken when using this command; in general, it should only be used as part of a calibration sequence. The safer movement commands (described below) should be used for normal motor movement.

Syntax: **!Module-Id 1 [motor][direction]**

Motor can be either A or B; direction **must** be specified as + (position step number increases; slit moves outward), or – (slit moves inward, step number decreases).

Returns:	`%Module-Id OK Xpos Ypos DONE;`	if successful
	`%Module-Id BUSY;`	if motors are already moving

Example: **! XIAHSC-B-0037 1 A+**

Motor A moves one step CCW

```
%XIAHSC-B-0037 OK 401 400 DONE;
```

6.3.3 A: Alias

Allows the assignment of a unique user-specified module alias, which can be used instead of the serial number to address a module. Only one alias per module is allowed; the alias can be up to 24 characters in length. Note that there can be no imbedded spaces. The alias can be cleared by using module alias = ‘-’.

Syntax: **!Module-Id A [Module-Alias]**

Returns: `%Module-Id OK [Module-Alias] DONE;` if successful
`%Module-Id BUSY;` if motors are moving (alias not changed)

Note: As in all commands, Module-Id can be either the module serial number or the previously assigned alias, so you can use the old alias while assigning a new alias.

Example: **! XIAHSC-B-0037 A Primary-Vertical-Slit**
`%XIAHSC-B-0037 OK Primary-Vertical-Slit DONE;`

6.3.4 C: Close Slits

Close the slit aperture by the specified number of steps. If the number of steps is even, the aperture center remains fixed. If the number of steps is odd, one slit will move one step more than the other slit. The choice of which slit moves more is alternated so that there will be no accumulated drift. This is true in general - an arbitrary sequence of open and close commands will not produce any accumulated drift of the center position.

Syntax: **!Module-Id C [∆ opening]**

Returns: `%Module-Id OK;` immediately if movement is allowed
`%Module-Id ERROR;` if movement not allowed or command error
`%Module-Id BUSY;` if motors are already moving (close command is aborted)

`%Module-Id Apos Bpos DONE;` after motion is complete

Example: **! Primary-Vertical-Slit C 100**
`%XIAHSC-B-0037 OK;`
… motors move
`%XIAHSC-B-0037 750 1000 DONE;`

6.3.5 I: Inquiry

Returns a summary of the current status of a module, including motor positions, limits, serial number and alias.

Syntax: **!Module-Id I**

Returns:

```
%Module-Id OK HSC v1.0 (c) XIA 1998 All Rights Reserved
SERIAL: Module-Id
ALIAS: [Module Alias]
Motor A @ nnnnn (steps)
Motor B @ mmmmm (steps)
Limits Enabled: [YES or NO]
Calibrated: [YES or NO]
Motor A Limits: 0 to nnnnn
Motor B Limits: 0 to nnnnn
DONE;
```

6.3.6 K: Kill Movement

This command immediately stops all motor movement. In order to stop all modules at once, use the ALL module address, described above.

Syntax: **!Module-Id K**

Returns: Normal completion response from whatever movement command was interrupted.

6.3.7 M: Move (absolute for Version 1.2 or less; absolute or relative for version 1.3 and later)

For code version 1.2 or earlier, this command instructs the controller to move the slits to absolute positions specified in the command line; positions must be specified for both motor A and motor B. Recall that all positions must be specified in terms of steps.

Syntax: **!Module-Id M [Apos] [Bpos]**

Returns:

```
%Module-Id OK;              immediately if movement is allowed
%Module-Id ERROR;           if movement not allowed or command error
%Module-Id BUSY;            if controller is busy

%Module-Id Apos Bpos DONE;      after motion is complete
```

Example: **! Primary-Vertical-Slit M 1000 1500**

```
%XIAHSC-B-0037 OK;
… motors move
%XIAHSC-B-0037 1000 1500 DONE;
```

For code version 1.3 or later, the **M** command also supports relative and null motion. The '=' character specifies that the corresponding motor should not be moved. A '+' or '-' character indicates that the move will be relative, where the following number gives the number of steps to move. The '+' character is used to specify counter-clockwise motion (the absolute step number increases), and the '- ' character specifies clockwise motion (the absolute step number decreases). Any combination of absolute, relative and null motion can be combined in the same command. A space must separate the argument controlling motion for motor A and the argument specifying the motion for motor B.

Syntax for Relative motion:

!Module-Id M [+ or -][NstepsA] [+ or -][NstepsB]

Examples: **!Primary-Vertical-Slit M = -500**

```
%XIAHSC-B-0037 OK;
… motor B moves 500 steps CW
%XIAHSC-B-0037 1000 1000 DONE;
```

!Primary-Vertical-Slit M 2000 +100

```
%XIAHSC-B-0037 OK;
… motor A moves to 2000 absolute, motor B moves 100 steps CCW
%XIAHSC-B-0037 2000 1100 DONE;
```

!Primary-Vertical-Slit M 750 =

```
%XIAHSC-B-0037 OK;
… motor A moves to 750 absolute, motor B does not move
%XIAHSC-B-0037 750 1100 DONE;
```

6.3.8 O: Open Slits

Open the slit aperture by the specified number of steps. If the number of steps is even, the aperture center remains fixed. If the number of steps is odd, one slit will move one step more than the other; the choice of which slit moves more is alternated so that there will be no accumulated drift. Note that an Open command followed immediately by a Close command with the same argument will always return the slits to their position prior to the Open command.

Syntax: **!Module-Id O [Δ opening]**

Returns:

	`%Module-Id OK;`	immediately if movement is allowed
	`%Module-Id ERROR;`	if movement not allowed or command error
	`%Module-Id BUSY;`	if controller is busy
	`%Module-Id Apos Bpos DONE;`	after motion is complete

Example:

```
! Primary-Vertical-Slit O 100
%XIAHSC-B-0037 OK;
… motors move
%XIAHSC-B-0037 850 1100 DONE;
```

6.3.9 P: Position Inquiry

This command returns the current position of the two motors. It also provides a convenient way of polling the device to see if the motors are moving.

Syntax: **!Module-Id P**

Returns:

	`%Module-Id Apos Bpos DONE;`	if motors not moving
	`%Module-Id BUSY;`	if motors are moving

Example:

```
! Primary-Vertical-Slit P
%XIAHSC-B-0037 850 1100 DONE;
! Primary-Vertical-Slit M 2000 2000
%XIAHSC-B-0037 OK;
… motors are moving
! Primary-Vertical-Slit P
%XIAHSC-B-0037 BUSY;
%XIAHSC-B-0037 2000 2000 DONE;
… motors are no longer moving
! Primary-Vertical-Slit P
%XIAHSC-B-0037 2000 2000 DONE;
```

6.3.10 R: Read internal memory

This command allows the user to directly read many internal memory variables used by the controller; see the memory map (Table 2) for a list of available data.

Syntax: **!Module-Id R [Index of variable in memory]**

Returns: `%Module-Id OK [Variable value] DONE;` if successful
`%Module-Id ERROR;` bad index

Example: **! Primary-Vertical-Slit R 1**
`%XIAHSC-B-0037 OK 4400 DONE;` returns upper motion limit

6.3.11 S: Slide Slits

This command moves both slits in the same direction by the specified amount; the aperture size is unchanged. The direction must be specified, using either a '+' or a '–' as the first character of the command argument. Choosing the '+' direction increases the absolute position of slit A and decreases the absolute B position; the '-' direction does just the opposite.

Syntax: **!Module-Id S [+/-][Δposition]**

Returns: `%Module-Id OK;` immediately if movement is allowed
`%Module-Id ERROR;` if movement not allowed or command error
`%Module-Id BUSY;` if controller is busy

`%Module-Id Apos Bpos DONE;` after motion is complete

Example: **! Primary-Vertical-Slit P**
`%XIAHSC-B-0037 OK 650 1100 DONE;`
! Primary-Vertical-Slit S +100
`%XIAHSC-B-0037 OK;`
… motors move
`%XIAHSC-B-0037 750 1000 DONE;`

Note that Apos + Bpos (related to the slit gap) remains constant.

6.3.12 T: Test Module

This command puts the HSC module into Test mode. In this mode, the LED's are all turned on, and the motors respond to the buttons (as long as the unit is calibrated). In addition to moving the motors when the buttons are pressed, the controller writes the identity of the button that was pressed to the RS-232 transmit line to show that the controller detected the button being pressed as well as to test the RS-232 communications. A 20-second period with no buttons pressed ends Test mode.

Syntax: **!Module-Id T**

Returns: `%Module-Id OK;` if successful
`%Module-Id ERROR;` if unsuccessful

`%Module-Id TESTMODE DONE;` after Test mode is complete

Example: **! Primary-Vertical-Slit T**
`%XIAHSC-B-0037 OK;`
… press each button once:
`%XIAHSC-B-0037 A-CW;`
`%XIAHSC-B-0037 A-CCW;`
`%XIAHSC-B-0037 B-CW;`
`%XIAHSC-B-0037 B-CCW;`

… don't press a button for 20 seconds:

```
%XIAHSC-B-0037 TESTMODE DONE;
```

6.3.13 W: Write to internal memory

This command allows the user to directly modify many internal variables used by the controller; see the memory map (Table 2) for a list of variables. Note that some variables are read-only; check the visibility column to see what variables can be modified and which are read-only.

Syntax: **!Module-Id W [Index of variable in memory] [New Value]**

Returns:

```
%Module-Id OK [Old value] [New value] DONE;        if successful
%Module-Id ERROR;                             bad index or value, read-only
```

Example: **! Primary-Vertical-Slit W 1 4000**

```
%XIAHSC-B-0037 OK 4400 4000 DONE;       changes upper motion limit
```

7 Error Codes and Descriptions

The HSC-1 reports any errors using error codes, and optionally with brief text describing error. The error codes are described below.

7.1 Error 0: Missing Command

This error is reported when an HSC-1 module receives a valid address but no command follows. The address is followed by at least one space then a carriage return, with no non-space characters between the module id and the carriage return. No error is reported if the module id is immediately followed by a carriage return, since the controller uses a space to define the end of the module id.

7.2 Error 1: Unrecognized Command

This error is reported when the first non-space character after the module id is not a valid command character, as listed in Table 4.

7.3 Error 2: Input Buffer Overflow

This error indicates that the command was too long to be stored in the input text buffer. The buffer is long enough to hold any valid command, so this error indicates an invalid command (or perhaps a problem with the serial connection such that the carriage return that terminates the command is not recognized by the controller). The input buffer is 32 characters deep, and stores the whole command line, starting at the command character. Note that this error is reported as soon as the buffer length is exceeded, not after the terminating carriage return.

7.4 Error 3: No new Alias given

This indicates that a valid alias command was received, but no new alias was given (there were no non-space characters between the '**A**' command and the terminating carriage return).

7.5 Error 4: Alias too long

This error is reported when the new alias given in the '**A**' command exceeds the maximum alias length of 24 characters. The alias is left unchanged.

7.6 Error 5: Invalid Field Parameter

This error is reported in response to either a read ('**R**') or write ('**W**') command, and indicates the field argument (that is, the memory map index) is invalid. This means that either the specified number is not a valid index (any integer from 1 to 14), or that the argument could not be interpreted as an integer (it contains a non-numeric character).

7.7 Error 6: Value Out of Range

This error is reported in response the a write ('**W**') command, and indicates that the new value for the specified parameter is too large to be stored in the memory space allocated for that variable. See Table 2 for the allowed ranges for each variable.

7.8 Error 7: Parameter is read-only

This error is returned in response to an attempt to modify a read-only variable. See Table 2 to determine which variables can be modified by the user.

7.9 Error 8: Invalid/Missing argument

This error indicates that an argument is invalid, or that a required argument is missing (for example, if only one motor position is specified in the Move ('**M**') command).

7.10 Error 9: No Movement Required

This error is returned by any of the movement commands ('**M**', '**O**', '**C**' or '**S**') when no movement is required; that is, the specified position is the same as the current position.

Note: for code versions 1.1 and greater (see the introductory banner produced when the power is first turned on, or the first line of the Inquiry response), this condition is not reported as an error.

7.11 Error 10: Uncalibrated: no motion allowed

This error is returned by any of the movement commands if the HSC-1 is not calibrated (that is, the controller does not know the position of the slits). No motion is allowed when the HSC-1 is not calibrated, to avoid slit collisions or running the slits off their screws (see the single-step '**1**' command for the exception to this rule).

7.12 Error 11: Motion out of range

If motion limits are enabled, this error is returned by any movement command when the desired new position either results in a slit collision or moves one or both slits beyond their outer range limit.

7.13 Error 12: Invalid or missing direction character

This error is reported in response to either the slide (‘**S**’) or the single-step (‘**1**’) command when no direction is specified. The movement direction must be specified using either a ‘+’ or ‘-’ character; see Section 6 for more details.

7.14 Error 13: Invalid Motor Specified

This error is reported by the single-step (‘**1**’) command when the character specifying which motor to move is neither an ‘A’ or a ‘B’.