

---

## Chapter 5. PROGRAMMING

---

### Chapter Objectives

This chapter provides complete information on programming the positioner to perform the motion control functions you require.

---

### Communicating with the Positioner

The positioner is designed to work in one of two modes:

- a) The human interactive mode
- b) The computer host mode

The command EX1 enables the first of these and the command EX0 the second. In the human interactive mode, the positioner will respond to a dumb terminal in an intelligent, user friendly manner and provide the user with help facilities and error messages if errors occur. In the host computer mode all unsolicited output and help information is suppressed so that communication may be handled by software resident in the host computer.

If a computer is to be used for the terminal function, terminal emulation software will be required. The X-Ware program available from Digiplan for use with IBM PC's and compatibles is suitable for this purpose. X-Ware is supplied free of charge.

---

### Individual Commands

An individual positioner command controls a single parameter, function, or action such as acceleration, velocity, position, time delay, pause, loop, go, etc. There are two classes of individual commands, Immediate and Buffered.

Individual commands are variable in length. They can consist of one or more letters with a delimiter (carriage return or space character) and one or more letters and numbers with a delimiter. Each command is entered as a character/delimiter combination. Some commands include a sign ( $\pm$ ) to denote direction of motion. The number of characters used depends on the type of command entered.

Typical commands have the form:

```
1E
S
A10
V3
D46000
```

When two or more individual commands are entered on the same line, they are separated by spaces, and multiple command entries will be displayed on a single line of your terminal screen.

The example below shows a set of individual command entries with space delimiters on the same line:

```
MN A10 V2 D25000 L10 G N
```

If spaces are used as the delimiter and a large number of multiple command entries are made, you could exceed the ability of your terminal to display characters as a single line (80 characters per line is a typical value). When the carriage return is used as a delimiter, the cursor returns to the beginning of the next line.

**NOTE:** The Positioner will normally echo carriage return with carriage return line feed in terminal mode. If you get double line spacing you need to turn off the auto line feed facility on your terminal. In computer mode (EX0) only the characters received are echoed.

The example that follows shows the effect of using carriage return as the delimiter when the terminal is not in auto line feed.

```
MN  
A10  
V2  
D25000  
L10  
G
```

---

**Immediate  
Commands**

Commands that are identified as "Immediate" are executed immediately on receipt and will take priority over whatever operation is in progress. These commands include various Stop commands that clear the command buffer, and various Status Request commands that have no effect on the command buffer.

---

**Buffered  
Commands**

Commands that are identified as "Buffered" are received by the positioner and stored (in the command buffer) if the positioner is not free to execute them. Stored commands are executed as soon as the positioner gets to them, in the order that they were received.

---

**Buffer  
Capacity**

Any combination of individual commands and command groups can be entered in the buffer until the total number of characters currently stored (including the delimiters) equals 2000. The positioner uses a first-in-first-out serial buffer. As the commands are read from the buffer, additional commands can be entered to replace them. Therefore, the possibility exists that a command set could actually consist of more than 2000 characters. Examples of command sets that might be used for different applications are presented later.

The sequence buffer is different from the command buffer in that it can be copied to battery backed up RAM using the SV command, while the command buffer is not battery backed up, so its contents are lost on power down.

---

### **Multiple Positioner Commands**

When multiple positioners are on the communications line, commands like the previous example are executed by all positioners on the line. To send commands to a single positioner, the device address of that positioner should be put in front of the instruction. The device address is set using jumper links on the individual positioner.

Example:

Three positioners are on an RS232C chain. They are sent the following commands:

```
MN A10 V10 1D25000 2D50000 3T1 3D100000 G
```

Unit 1 moves 25000 steps, unit 2 moves 50000 steps, and unit 3 waits 1 second and moves 100000 steps. All three use the same rates.

**NOTE:** because multiple positioners receive characters in parallel, universal commands are always echoed by the last addressed axis (not by all positioners). An addressed axis continues to echo characters up to the next address i.e. the "2" of 2D50000 of the above example is echoed by axis 2 whereas 1D250000 and the space is echoed by axis 1. After power-up, axis 1 is the default axis.

During sequence download to one axis in a chain, all of the other axes must have their communication switched off by use of the F command. The alternative is to make each command in the sequence device specific by adding the device address.

---

### **Programming Modes**

#### **Normal Mode**

The command MN selects normal mode. In this mode moves are set by the D command. Within normal mode there are two options, absolute mode and incremental mode:-

---

#### **Absolute Mode**

The command MPA puts the positioner into absolute position mode. Distances entered using the D command are interpreted as absolute positions relative to absolute zero

Example:

|       |   |
|-------|---|
| MPA   | Sets absolute mode                        |
| PZ    | Sets current position to be absolute zero |
| D4000 | Sets move to absolute position 4000       |
| G     | Execute move                              |
| D2500 | Sets move to absolute position 2500       |
| G     | Executes move of 1500 steps CCW           |

---

**Incremental Mode**

The command MPI puts the positioner into incremental positioning mode. D command distances are interpreted as the number of steps to move from the current position.

Example:

|         |                                  |
|---------|----------------------------------|
| 1MPI    | Sets incremental mode            |
| 1DPA    | Response=1500 (current position) |
| 1D4000  | Sets move to be 4000 steps CW    |
| 1G      | Executes move to position 5500   |
| 1D-3000 | Sets move to 3000 steps CCW      |
| 1G      | Executes move to position 2500   |

---

**Continuous Mode**

The command MC selects continuous mode in which the motor runs continuously at the specified velocity until stopped or a new velocity is programmed. Distance data is ignored.

Example:

|       |   |
|-------|---|
| MC    | Sets continuous mode                          |
| D4000 | Ignored in continuous mode                    |
| A50   | Sets acceleration to 50 revs/sec <sup>2</sup> |
| V20   | Sets velocity to 20 revs/sec                  |
| G     | Motor runs at 20 revs/sec continuously        |
| V10   | Motor speed changes to 10 revs/sec            |
| S     | Motor stops                                   |

---

**Speed Change Mode**

The MQ command allows speed changes to be made during a preset move. The speed changes can be triggered at a set distance using the TRD command or by an input using the TRE command.

Example:

|         |   |
|---------|---|
| 3MQ     | Set MQ mode for axis 3                              |
| 3A50    | Set axis 3 acceleration to 50 revs/sec <sup>2</sup> |
| V15     | Set velocity to 15 revs/sec                         |
| D8000   | Set move distance to 8000 steps                     |
| G       | Start move  |
| TRD3000 | Go to next command at D = 3000 steps                |
| V5      | Change velocity to 5 revs/sec                       |
| TRE_X1  | Go to next command when Input 2 goes high           |
| V2      | Change velocity to 2 revs/sec                       |

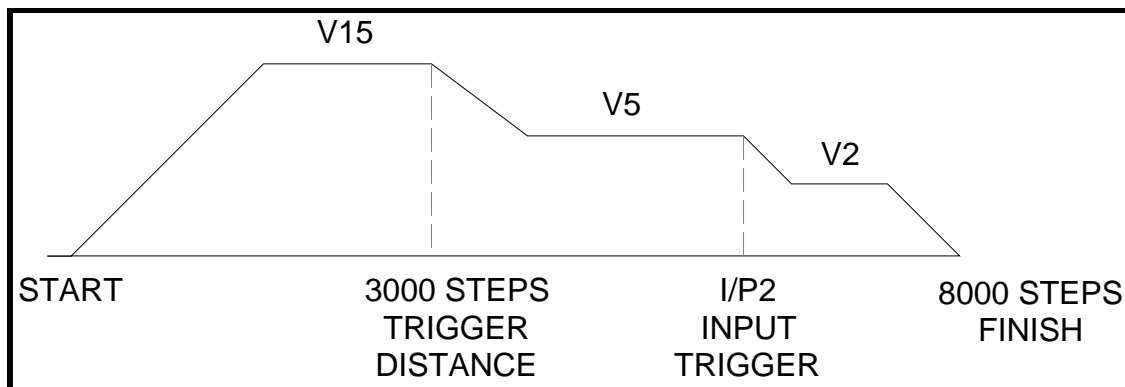


Figure 5-1. Speed Change Mode

## Command Types

### Start Move Commands

The positioner reads and stores individual commands in the order they are entered. Each command is read and executed before the next command is read. The GO command "G" is the principal buffered command that initiates shaft motion. When the buffer reads a GO command, the motor will then move in accordance with the move parameters that reside in the positioner at the time the Go command is read.

When the Go command is read, the positioner will incorporate any motion parameter command that was encountered prior to the Go command. For example, the "A" command would be combined with the Go command to change the acceleration value as follows:

A123 G

This command instructs the Positioner to change the acceleration value to 123 rev/sec/sec and then execute the move using the last mode, velocity, and distance values that were previously entered.

When the RS232C positioner is initially activated, a series of commands of the form " MN An Vn Dn G " is required. These command sets up the initial operating conditions for the positioner. After the positioner has been initialised, single move parameter commands can be used.

If only "G " is entered, the positioner will repeat the previous motion pattern. For example, a string of Go commands:

G G G G

would instruct the positioner to move the motor the same way four times.

There is no need to re-enter all the shaft motion commands to change one of the variables. If one or more of the motion parameters is changed by a command entry and the Go command is then entered, the prior pattern will be repeated using the new motion parameter(s) that were entered prior to the latest GO command.

For example:

A14 G V2.6 G D-27634 G

would change acceleration to 14 and move, then change velocity to 2.6 and move with 14 as the acceleration value, then change position and move again using 14 and 2.6 as the acceleration and velocity values.

The CONTINUE command "C " is used following execution of a PAUSE command, "PS " or "U ". CONTINUE will allow execution of the next command waiting in the command buffer (if any command is stored).

---

### Loop Commands

The LOOP command allows a cycle to be repeated continuously "L" or a given number of times "Ln" up to 65535. The END-OF-LOOP command "N" indicates where the loop ends. The END-OF-LOOP command can be used to indicate that the positioner should proceed with further commands after the designated numbers of loops have been executed, or in combination with the "Y" command to indicate where execution is to stop. The "U" command may be used to temporarily halt Loop execution, the C command will then cause the loop to resume execution.

### Stop Move Commands

The STOP MOVE commands are presented in order of severity of response. SOFTWARE RESET "Z" command is the most abrupt and

severe stop command you can use. The STOP AT END OF CURRENT LOOP command "Y" is the least abrupt and severe stop command.

**Stop at End of Current Loop Y**

Also the 'stop sequence' command.

This command will not halt processing until command processing reaches the character N at the end of the command loop. At that time, the Positioner will execute the next command in the buffer after N, if any. The command loop cannot be restarted without re-entering the commands.

**Controlled Stop command S**

In the PRESET or CONTINUOUS mode this command will decelerate the motor to a stop at the last used acceleration rate.

An "S " command will always cause a deceleration to velocity zero at the last used acceleration.

The "S " command clears any remaining commands in the command buffer unless prevented from doing so via the SSH command.

**NOTE:** Normally, the motor is decelerated to a stop at the same rate as it was accelerated. A different deceleration rate may be programmed in CONTINUOUS mode only by calling for a velocity of zero rev/sec with a new acceleration and executing a V0 rather than an "S " command.

Example:

```
" MC A1 V10 G .... A100 V0 G "
```

**Controlled Stop command LS**

The LS or 'Limit Stop' command will decelerate the motor to a stop at the deceleration rate defined by the LA or 'Limit Acceleration' command. This will guarantee a quick stop for any move.

**Kill Command K**

This command stops positioner commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

**Software Power-on Reset Z**

The "Z " command is equivalent to cycling the AC power to the Positioner; that is, it disables the communications interface and returns all internal settings to their power-on values. The command buffer is cleared. Like the "K " command, "Z " causes an immediate cessation of output pulses to the motor. Z also de-energises and resets the drive ready for re-initialisation at power up.

**NOTE:** When the "Z " command is used, the Positioner is busy for up to 2 sec and will ignore any commands.

---

### Pause Commands

There are four types of pause commands available as follows:-

Pause and wait for continue command C to continue. A common use of the Buffered Pause PS is to hold execution of individual commands in a Command Loop until the entire loop has been loaded.

**Tnnn** Pause in processing commands for a given number of seconds and then continue.

**U** Hold (pause) now and wait for continue. This is an immediate command.

**TRE** Pause in processing commands until the designated TRIGGER input pattern is present.

Triggers are used to synchronise positioner operations with external events. They can be used to implement a "handshaking" function with other devices. See command details for exact syntax.

---

### Status Request Commands

All status request commands result in data being returned to the controller from the positioner. To prevent multiple positioners from all responding at once, the status request commands are given the classification "Device Specific" meaning that the device address of the responding positioner must be placed in front of the command. No positioner will execute a Device Specific command without its device number.

The use of status request commands must be conducted in an orderly fashion. Commands should only be issued when the host is ready to read the response. New commands should not be sent until the response is received. In particular, after a buffered status command, an immediate status command should not be sent until the response has been received by the host. If this procedure is not followed, the command responses will be intertwined, rendering the information useless.

There are two status commands that can be used to request position.

The DPA Command will report how many steps the motor has moved relative to the absolute zero position. This position is calculated by summing the total number of moves commanded since the positioner was at the absolute zero position. If position reports are needed that are relative to the beginning of each individual move, the PR command should be used.

---

### **Drive Energise/De- energise Function**

Energising and de-energising the drive is subject to the commands given and error status as follows:

The drive will energise if:-

The ST0 or ON command is given when the ESTOP, WATCHDOG, FOLLOWING ERROR and EXCESS TORQUE DEMAND signals are inactive.

The drive will de-energise if :-

The ST1 or OFF command is given.

The ESTOP or WATCHDOG signals become operative.

A FOLLOWING ERROR or an EXCESS TORQUE demand occurs.

The de-energising signal must be restored, then the ST0 or ON command must be given to re-energise the drive.

---

### **Homing Function**

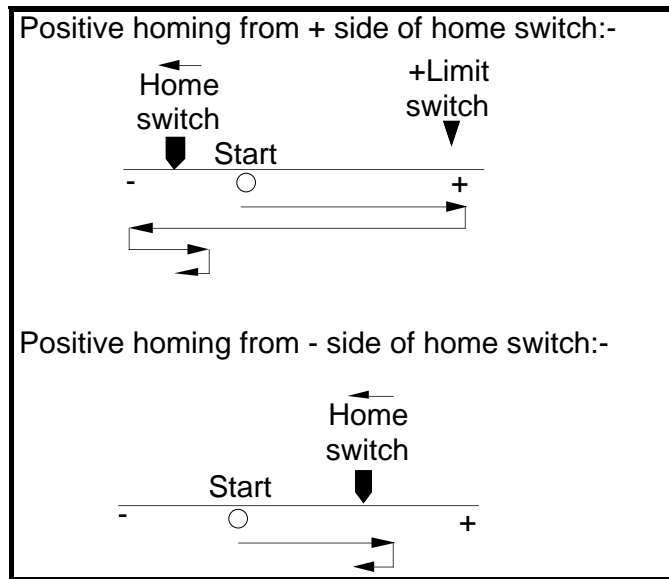
The GH command causes the motor to rotate in the direction and at the speed specified until the home position is reached. In practice, it continues a little beyond this point but you can locate the home position accurately by then using the commands:

MPA D0 G

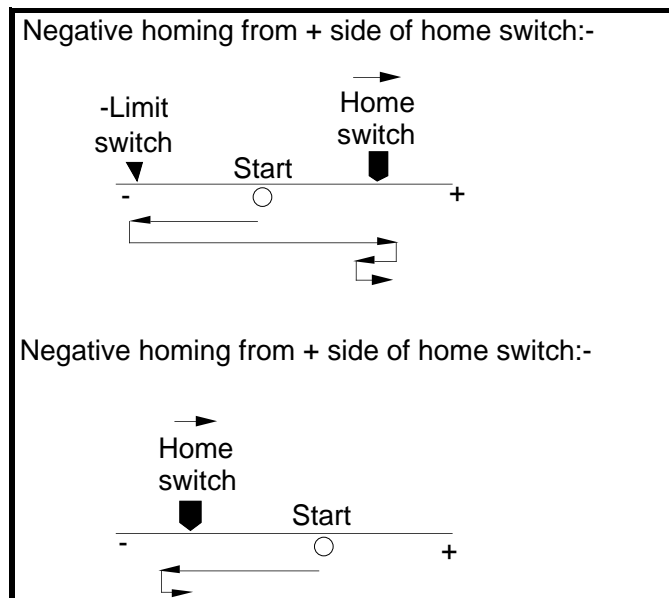
(The GH command always makes the home position zero)

The SP command can be used after homing to change the accurate home position from zero if required.

If the home position is not found in the direction given in the GH command, the limit switch position will be reached and the switch will be operated. The motor will then reverse and seek the home position in the opposite direction. For recognised home detection, the home switch must always be activated from the direction specified in the command (shown by the arrow in Figure 5-2), so as illustrated in this figure, a limit switch reversal will take place if the direction given in the GH command is away from the home position (GH+ in the illustration).



**Figure 5-2. Positive Homing**



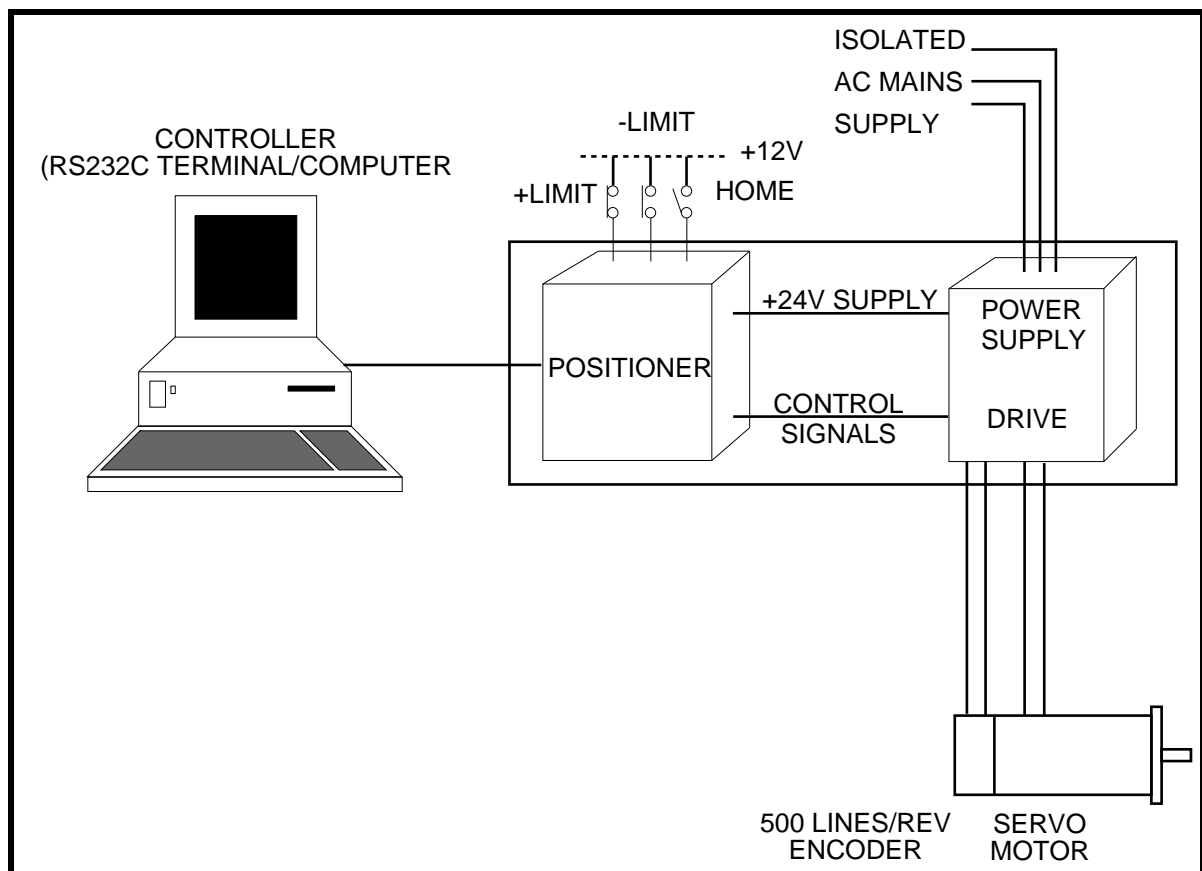
**Figure 5-3. Negative Homing**

For servo systems fitted with an incremental encoder with an index track, option OSA enables homing to an encoder index mark within the home proximity range, as opposed to the home switch edge. The mark is always approached from the specified direction. Using an index mark to determine the stopping point results in the home position remaining fixed even if there are small variations in the operating point of the home switch. The final home position will be the leading edge of the first index pulse following detection of the home switch edge.

## Basic Programming Guide

This section uses examples of simple programs as an introduction to programming. If you are unfamiliar with the system, after reading Chapter 1 a study of this section will give you an insight to the usage of the command language.

The system configuration is a brushless servo motor with a 500 line/rev encoder. 4X decoding is used and the user resolution has been set to 200 steps/rev. Tuning the servo loop is dealt with in the section headed "Servo Tuning", so we will assume that it has been completed. The positioner jumper links have been set for unit 1 addressing. The motor starting position is 400 steps CW from the home position.



**Figure 5-4. System Used in Example Programs**

**Example 1** In this simple example the requirement is to cause the motor to perform two revolutions in a CW direction and stop.

The following commands in the order shown will fulfil the requirement:-

| <u>Command</u> | <u>Purpose</u>                        |
|----------------|---------------------------------------|
| <b>MN</b>      | Set move mode to normal               |
| <b>1A10</b>    | Set acceleration motor 1 = 10         |
| <b>1V5</b>     | Set velocity motor 1 = 5              |
| <b>1D400</b>   | Set move distance motor 1 = CW 2 revs |
| <b>1G</b>      | Start move - motor 1 moves 2 revs CW  |

**Description** Commands 2 - 5 cause motor 1 to accelerate at 10 revs/sec<sup>2</sup> to a velocity of 5 RPM, travelling CW for 400 steps (2 revs) at the G command.

**Example 2** This example causes axis 1 to go to the home position, then turn 5 revolutions CCW at 10 revs/sec and set the resulting position as absolute zero. The motor finally turns 8 revolutions CW to finish at absolute position 1600.

| <u>Command</u> | <u>Purpose</u>  |
|----------------|---|
| <b>ST1</b>     | Energise the drive                                      |
| <b>A10</b>     | Set acceleration to 10 revs/sec <sup>2</sup>            |
| <b>V10</b>     | Set velocity to 10 revs/sec                             |
| <b>GH-2</b>    | Send axis 1 to go home (2 revs/sec CCW)                 |
| <b>1D-1000</b> | Set axis 1 to go 5 revs CCW                             |
| <b>G</b>       | The axis moves to the required absolute zero position   |
| <b>MPA</b>     | Set absolute position mode                              |
| <b>PZ</b>      | Set position to absolute zero                           |
| <b>1D1600</b>  | Set move to abs. position 1600 (8 revs CW)              |
| <b>G</b>       | Move to position 1600 (1 rev CW from starting position) |

**Description** Commands 2 - 6 cause the motor to go to the home position and then turn 5 revolutions CCW. Command 7 sets the absolute positioning mode and Command 8 sets the resulting position as absolute zero. Commands 9 and 10 cause the motor to turn 8 revolutions CW to absolute position 1600 which is 1 revolution CW from its starting position.

## Sequences

**Introduction** A sequence is a series of commands that is executed in order whenever the sequence is run. Immediate commands cannot be stored in a sequence since they cannot be stored in the command buffer. The only "sequence legal" commands are buffered

commands.

The positioner has a section of non-volatile RAM totalling 6400 characters allocated for storing up to 63 sequences. The sequence buffers are of variable length, so they can store long sequences of up to 6400 characters or several shorter ones, provided that the total length of all sequences does not exceed the allocated 6400 character space.

During normal operation a command can specify the execution of any sequence stored in memory.

Sequence identifiers: number from 1 - 63.

When a sequence is programmed, the positioner automatically calculates a checksum for the sequence and stores the length of the sequence and its checksum in an internal directory.

The command XSS can be used to verify the existence of a sequence.

If a power-on sequence is used, the sequence and its sequence number are stored in the non-volatile RAM. A power-on sequence execution failure will be indicated by flashing the LED.

---

**Programming  
Sequences**

Sequences are programmed by first sending a special start command XD, which clears the command buffer. All subsequent commands are stored in the command buffer (but not executed) until an end of sequence command (XT) is received. Any condition which prevents proper recording of the sequence is saved and can be accessed with the status command XSD. The programmed sequence can be tested with the run sequence command XR, which executes any specified sequence.

The XU command can be used to read back the entered sequence.

To begin the programming of a sequence, the XD command immediately followed by sequence identifier number (1 to 63) and a delimiter must be entered. An XT command ends the sequence. The save command SV should be used to store the sequence into the positioner's non-volatile memory.

The commands entered after the XD command and before the XT command will be executed in the order in which they were entered when the sequence is run.

Example:

**NOTE:** The ">" prompt is replaced by the "\$" prompt when in sequence edit mode.

```

XD1      Begin definition of sequence No.1
A10      Set acceleration to 10 revs/sec/sec
V10      Set velocity to 10 revs/sec
D4000    Set move distance to 4000 steps
G        Go
H        Reverse next move
G        Go
XT       End of sequence

```

\* The definition of sequence No.1 is terminated by XT but the sequence is NOT stored in non-volatile memory until the save command SV is given.

**NOTE:** Remove LK1 on the positioner card (see Figure 4-3) to write-protect sequences.

Whenever the above sequence is run, the motor will turn 1 revolution CW and then 1 revolution CCW.

## Running Sequences

### By Command Input

A sequence can be run by entering the XR command immediately followed by a sequence identifier number (1 - 16) and a delimiter.

Example:

```
XR3[space] runs sequence No.3
```

### Standalone Operation

Stored sequences may be automatically executed when you power up the system or executed by remote switches.

### By Hardware Inputs

A sequence selected as a binary pattern set by input switches can be run in response to a trigger on Input 4 if SSF1 is set.

A sequence selected as a binary pattern, set by input switches, can be run in response to a trigger on Input 4 if SSF1 is set.

Selection is by Inputs 1-3 if OSD0 is set. (1-7)  
 Selection is by Inputs 1-3 and Input 7 if OSD1 is set. (1-15)  
 Selection is by Inputs 1-3 and 5-7 if SSI1 is set. (1-63)

Table 5-1 shows how the binary sequence number can be setup using inputs I1 to I7 (I4 being used as an input strobe line). D5 is the most significant binary bit and D0 the least. Sequences 1-7, 1-15, or 1-63 can be selected depending upon what options are set using SSI, OSD and SSF.

| Options set |        |      | I7   | I6   | I5   | I4     | I3   | I2   | I1   |
|-------------|--------|------|------|------|------|--------|------|------|------|
| SSI0        | OSD0   | SSF0 | User | User | User | User   | User | User | User |
| SSI0        | OSD0   | SSF1 | User | User | User | Strobe | D2   | D1   | D0   |
| SSI0        | OSD1   | SSF1 | D3   | User | User | Strobe | D2   | D1   | D0   |
| SSI1        | OSD0/1 | SSF1 | D5   | D4   | D3   | Strobe | D2   | D1   | D0   |

**Table 5-1. Sequence Selection**

Example:

Set SSF1 and set Inputs 1-3 and Input 7 as follows:

Input 1 = 1  
 Input 2 = 0  
 Input 3 = 0  
 Input 7 = 1

On an Input 4 transition from low to high level, sequence number 9 will be executed if OSD1 and SSF1 have been set. The Input 4 trigger will not be accepted if the axis is busy. When a PLC is used, it is recommended that an output (O3 for example) is set at the beginning of a sequence to indicate to the PLC that the axis is busy. A command could then be added at the end of the sequence to reset the output and indicate to the PLC that the axis is ready to receive a new trigger.

**Power On Sequence**

One of the stored sequences can be designated to be executed when the device is powered up. The sequence to be executed can be selected by prior RS232C command (XP1-16). The single sequence specified is executed once; control then passes to the normal command processor loop. Alternatively, power-on execution can be disabled with either the XP0 or XZ commands.

It is possible to have the positioner execute a sequence on power up by defining the commands within a sequence using the XD, XT and XP commands. To define sequence No.1 the operator enters:

XP1 XD1 MN A10 V5 D40000 G XT

In this example the operator would not power the unit on until he was ready for the sequence to begin. The XP1 command enables a single run of sequence No.1 after power on. The XD1 command signifies the start of the definition of sequence No.1 and the XT command signifies the end of the sequence No.1 definition. When the operator applies power to the positioner, sequence number 1 will be executed (the motor will turn 10 revolutions in the positive direction). The sequence can be stopped using the S command or, if necessary, by using the emergency stop (K command).

#### **Changing a Programmed Sequence**

Once a sequence is defined it cannot be redefined until it has been deleted. A new sequence cannot be downloaded over an existing one.

To change a sequence, it must be deleted from memory and then redefined as a new sequence using the same number.

First, the sequence must be deleted from memory using the XE command. It will not be deleted from back-up RAM unless the SV command is used after the XE.

Example:

XE1 erases sequence No.1

The changed sequence should then be entered. It will not be saved to non-volatile memory until the SV command is issued.

#### **Examining the Contents of a Sequence**

If the operator wishes to check the contents of the sequence, he would enter:

1XU1

This would cause the positioner to send the contents of sequence number 1 to the terminal's screen. The 1 preceding the XU command is the device address which must be present since the XU command is a "device specific" command. After issuing this command the following will be displayed on the terminal's screen:

MN A10 V5 D40000 G

This is the command string that is performed when sequence number 1 is executed.

If the operator does not wish to have the sequence execute on power up, it can be cancelled by issuing either the XP0 or XZ command.

Sequence no.1 can also be run manually using the XR command.

#### **Sequence Status Requests**

Sequence status commands include XC, XSD, XSP, XSR, XSS. The XC command can be used to verify that the memory in the back-up

RAM has not been corrupted in any way since last power down. The XC command will cause the positioner to send a checksum in the form of three decimal digits (000 - 255) to the terminal. The checksum may be recorded when sequences have been defined, so that it may be used for comparison later, such as each time the positioner is powered up.

---

### **Interactive operation with a PC/PLC**

To create a reliable, trouble free interactive RS232 communications control program between a PC/PLC and the serial interface you should note the following points :

- Use the X-WARE package for terminal emulation on an IBM PC or equivalent computer system.
- The interface should only be operated in computer mode to suppress error messages and extended text output.
- The program should operate the interface with a response check during critical processes, such as sequence download. The response check will need to detect :
  - No response at all
  - Incorrect characters
  - More characters than expected
  - Less characters than expected
  - Control characters such as EOF caused by errors
  - Correct response

The response expected should be either echoback, or character returned by interrogation commands, or the sequence of the two.

- If there is no check for echoback on each character, you should include a delay of up to 1ms between each character sent . This may require characters to be sent individually rather than as strings.
- You should not use the simple BASIC INPUT statement as this 'hangs' the link without time out if an error causes no carriage-return character to be received from the interface.
- Interpreted BASIC should not be used, as it is too slow for most applications.
- Interrupt error handling, such as the BASIC 'ON ERROR' statement, should be enabled.
- Commands such as SV, Z, RFS and RIFS take several seconds to execute. This also applies to status commands like DR and HELP. It is recommended you insert a suitable delay after these commands to allow them to fully execute their intended function

without corrupting immediately following commands, although the use of these commands in an interactive situation is unlikely.

- If problems arise the communications error responses actioned by the R command will help diagnosis. In some cases a recovery can be attempted by sending the command again (such as the R command itself). But this may not always be appropriate, for example LD3 altered to L3 would not be fully corrected by simply sending LD3 again.
- A routine that breaks some of these rules may appear to work satisfactorily, but is not necessarily safe. The worst case communications response condition may involve a number of coincidences, such as sending a status request just as the interface hits a limit, just as it starts decelerating or just as the servo is responding to a load change. Any motion control requirement explicitly takes priority over communications under these circumstances, and can therefore cause a communications problem unless you wait for echoback. Also different issues of software may have different execution time characteristics which can make a routine based on delays work at one software issue, and not another.

Accurate defensive programming is required if safety constraints are to be met and possible damage to hardware prevented.

---

## Command Structure

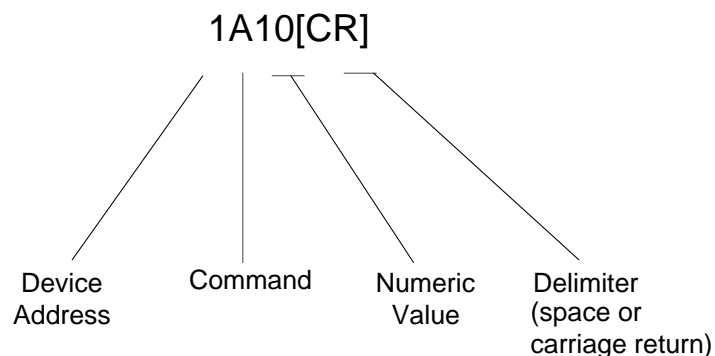
This section defines the basic format for positioner commands.

---

### Basic Format

All positioner commands are sequences of upper case characters of the form:

[device address] [command] [numeric value] [delimiter]



where:

**device address** - is a single digit from 1 to 8. It determines which device will receive the command. Its value will be between 1 and 8 since up to 8 positioners may be present in a system. If the device address is omitted, the command applies to all positioners.

**command** - up to four characters beginning with an uppercase ASCII letter

**numeric value** - a number defining the value to be programmed, such as the number of steps to move

**delimiter** - is a SPACE or CARRIAGE RETURN

All individual commands end in a delimiter that signifies that the command is complete. A delimiter serves the same function as the space between words in a sentence. The delimiter, which is part of the command, is a space character (entered with the keyboard space bar) or a carriage return [CR].

The format for each individual command is given by the **syntax**, for example:

`<a>GH<s>n`

`<a>` is the device address

`GH` is the command

`<s>` is the sign, indicating direction of travel

`n` is the numeric value, in this case the speed in revs/sec

Characters inside `<>` are optional. The direction is taken to be positive if no sign is included.

**Sign `<s>`** s = sign (+ or -)

The sign placed before the numeric value indicates the required direction of travel for a move. If the sign is omitted, a positive value is assumed.

**Numeric Value `n`** The value of `n` may be expressed as:

an Integer (a whole number such as 40)

a decimal number such as 10.25

a scientific E notation number such as +12.3840E-04 etc.

**NOTE:** All commands which take a value, such as a velocity command like 1V5, will return the current programmed value if none is sent. So sending 1V[CR] will result in the positioner responding with a message such as \*V = 5 revs/sec.

**Defaults**

Command parameters have default or factory setting values which are listed for each command. The value given is the general product default (corresponding to the commands RFS0 followed by RIFS). See commands RFS and RIFS for further information.

---

**Command Attributes**

All commands are either Immediate or Buffered and Universal or Device Specific, as explained below.

**Immediate (I):** Executed immediately upon receipt.

**Buffered (B):** Stored in the command buffer and executed in sequential order.

**Universal (U):** Commands that are intended for all positioners in the system. They do not require a device address. Any universal command can be made device specific by including a device address.

**Device Specific (D):** Commands that are intended for one positioner only and therefore require a device address for execution. All commands that request data to be transmitted back to the host are device specific. This includes position report backs and status requests. All 'Universal' commands can be made 'device specific' by including a device address, except the 'Z' (reset) command.

**Buffered/Immediate Commands**

All commands annotated with \* can be prefixed by the character B, in which case they become buffered rather than immediate commands.

**WARNING:** Although every effort is made to ensure that the default values of parameters are harmless and conservative, there can be problems when the unit is connected to non-standard servo drives or loads. Under these circumstances follow the tuning procedures starting with low servo gains as described.

## DETAILED COMMAND LIST

| <b>A</b>           | <b>Acceleration Rate</b>   |                        |   | <b>VALID FROM</b><br>Version 1.4           |
|--------------------|--|------------------------|---|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>           | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                          |
| <a>An              | Revs/sec <sup>2</sup>  | n = 0.030 -<br>999,999 | 10  | Buffered, Universal<br>Savable in sequence |
|                    |  | <b>EXECUTION TIME</b>  |   | <b>SEE ALSO</b> V, D                       |
| <b>Description</b> | Set the acceleration rate. The acceleration remains set until it is changed or the system is reset.    |                        |   |  |
|                    | If n = 0 there is no acceleration period. The maximum possible acceleration is used (start/stop move). |                        |   |  |
| <b>Example</b>     | <u>Command</u>   |                        | <u>Description</u>  |  |
|                    | 1A10   |                        | Sets acceleration of device 1 at 10 revs/sec <sup>2</sup>   |  |
|                    | A5   |                        | Sets acceleration of all devices to 5 revs/sec <sup>2</sup> |  |

| <b>B</b>           | <b>Buffer Status Request</b>   |                       |                | <b>VALID FROM</b><br>Version 1.4           |
|--------------------|--|-----------------------|----------------|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                          |
| aB                 | N/A  | N/A                   | N/A            | Immediate, Device Specific,<br>Never Saved |
|                    |  | <b>EXECUTION TIME</b> |                | <b>SEE ALSO</b> BS                         |
|                    | <b>RESPONSE TO</b> aB IS *B or *R  |                       |                |  |
| <b>Description</b> | The command requests the status of the command buffer. The response, *R[CR] or *B[CR], indicates if more or less than 10% of the command buffer is free. |                       |                |  |
|                    | *R = more than 10% of the buffer is free<br>*B = less than 10% of the buffer is free   |                       |                |  |
|                    | The command is normally sent after a long series of commands to ensure that room exists for more commands. The communications buffer is 2000 bytes long. |                       |                |  |

**Example**Command  
**1B**Response  
\*B (less than 10% of command buffer is free)**BS****Buffer Size Request****VALID FROM**  
Version 1.4**SYNTAX**

aBS

**UNITS**

Bytes

**RANGE**

0 - 2000

**DEFAULT**

N/A

**ATTRIBUTES**Immediate, Device Specific,  
Never Saved**EXECUTION TIME****SEE ALSO** B**RESPONSE TO** aBS IS \*n**Description**

Requests the remaining bytes available in the command buffer. The response, 4 decimal digits (0-2000) followed by a [CR], indicates the number of bytes remaining in the command buffer. When entering long strings of commands, check the buffer status to ensure that there is enough room in the buffer, otherwise commands will be lost. Each character (including delimiters) uses one byte.

**Example**Command  
**1BS**Response  
Response \*122[CR] (122 bytes remaining in the buffer)**C****Continue****VALID FROM**  
Version 1.4**SYNTAX**

&lt;a&gt;C

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**Immediate, Universal,  
Never Saved**EXECUTION TIME****SEE ALSO** PS, U**Description**

The continue command ends a pause state. It enables the interface to continue executing buffered commands after a pause has been initiated with the PS command or the U command. This command is useful when you want to transmit a string of commands before you actually execute them.

**Example**

| <u>Command</u> | <u>Description</u>                                  |
|----------------|---|
| <b>MC</b>      | Set continuous mode                                 |
| <b>A10</b>     | Set acceleration to 10 revs/sec <sup>2</sup>        |
| <b>V10</b>     | Set velocity to 10 revs/sec                         |
| <b>PS</b>      | Pause   |
| <b>G</b>       | G will not be recognised until PS is cleared with C |
| <b>T10</b>     | Delay for 10 secs                                   |
| <b>V0</b>      | Stop  |
| <b>C</b>       | Continue  |

|            |                                    |                                  |
|------------|------------------------------------|----------------------------------|
| <b>CAG</b> | <b>Configure Acceleration Gain</b> | <b>VALID FROM</b><br>Version 3.2 |
|------------|------------------------------------|----------------------------------|

| SYNTAX  | UNITS        | RANGE        | DEFAULT | ATTRIBUTES           |
|---------|--------------|--------------|---------|----------------------|
| <a>CAGn | Milliseconds | 1.0 to 32768 | 1.00    | Immediate, Universal |

**EXECUTION TIME**

**SEE ALSO** SIMM, CCS

**Description**

During following in scaling mode (SIM3, 4) the acceleration profile to the input stream velocity can be adjusted using the CAG command. CAG sets the time constant of a digital filter which profiles or smoothes the input pulse stream. In response to a step change in input frequency, the output will accelerate or decelerate to the new rate following an exponential profile. The CAG value represents the time in milliseconds to reach 67% of the final speed; 95% of the final speed is reached after 3 times the CAG value.

**Example**

| <u>Command</u> | <u>Description</u>                       |
|----------------|--|
| <b>ST0</b>     | Motor energised                          |
| <b>SIM3</b>    | Normal unreversed scaled following       |
| <b>CCS3</b>    | Clock and direction decode               |
| <b>CUR200</b>  | The input pulse stream resolution        |
| <b>CMR4000</b> | The system 7 motor resolution            |
| <b>CAG100</b>  | 100ms input filter controls acceleration |
| <b>1SV</b>     | Save, and power up in the following mode |

|            |                               |                                  |
|------------|-------------------------------|----------------------------------|
| <b>CCP</b> | <b>Configure Command Peak</b> | <b>VALID FROM</b><br>Version 3.0 |
|------------|-------------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE        | DEFAULT | ATTRIBUTES           |
|--------|-------|--------------|---------|----------------------|
| aCCPn  | N/A   | n = 0 - 1023 | 1023    | Immediate, Universal |

**Description** This command can be used to limit or clamp the maximum value of the output demand to a value less than the full range of -1023 to +1023. Since the drive is configured as a torque amplifier, CCP serves as a torque-limiting function.

**Example**

|                |   |
|----------------|---|
| <u>Command</u> | <u>Description</u>  |
| <b>1CCP102</b> | Limit the interface output to -102 to +102 i.e. 10% of full scale |

This clamping action is the last to take place before the control signal is output, therefore a command such as CCP0 would block any output signal to the drive.

|            |                                 |                                  |
|------------|---------------------------------|----------------------------------|
| <b>CCS</b> | <b>Configure Command Source</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|---------------------------------|----------------------------------|

|               |              |              |                |                           |
|---------------|--------------|--------------|----------------|---------------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>         |
| aCCSn         | N/A          | n = 0 - 3    | 0              | Buffered, Device Specific |

**EXECUTION TIME** **SEE ALSO** SIM

**Description** This command is used to configure the decode of the second axis encoder input when using following mode of operation as follows:-

- n = 0 Normal x4 decode
- n = 1 x2 decode
- n = 2 x1 decode
- n = 3 Clock and direction decode

**Example**

|                |                                       |
|----------------|---------------------------------------|
| <u>Command</u> | <u>Description</u>                    |
| <b>1CCS1</b>   | Set axis 1 to x2 decode               |
| <b>1SIM1</b>   | Select encoder following mode, axis 1 |

If an encoder to be tracked by axis 1 has 1000 lines per rev, it is decoded as 2000 steps per rev (x2 decode). The motor on axis 1 with 4000 steps per rev would move 1/2 revolution in response to a 2000 step move (1 revolution) of the tracked axis.

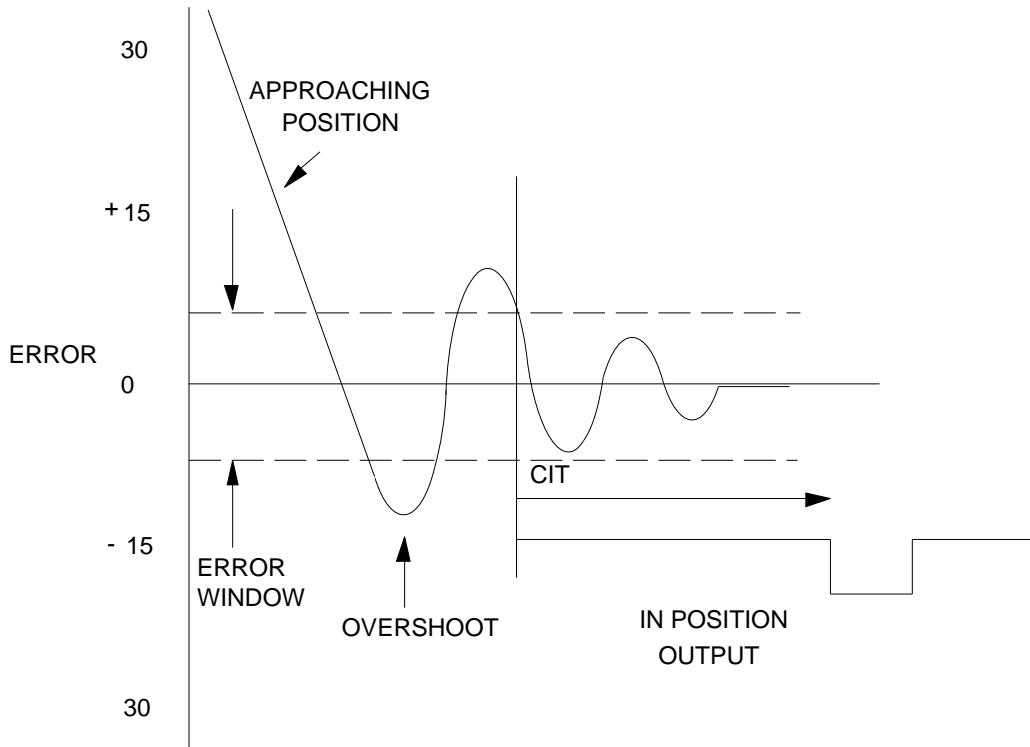
|             |                                     |                                  |
|-------------|-------------------------------------|----------------------------------|
| <b>*CEW</b> | <b>Configure In-Position Window</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|-------------------------------------|----------------------------------|

|               |              |                |                |                      |
|---------------|--------------|----------------|----------------|----------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b>   | <b>DEFAULT</b> | <b>ATTRIBUTES</b>    |
| <a>CEWn       | Steps        | n = 0 - 32,767 | 50             | Immediate, Universal |

**Description** This command, together with the CIT command, can be used to configure an in-position window, which in its turn can be used to

indicate that the preceding index has terminated. Output 02 can be configured with the SSC command to reflect the state of the in-position detector, thus allowing the user to trigger external hardware from the in-position condition. The in-position condition is met when:-

- a) The indexer algorithm has finished (no input position command)
- b) The CEW condition is met i.e. the following error is less than that specified by CEW
- c) The above condition has been true for the length of time specified by the CIT command



**Figure 5-5. Final Positioning**

**\*CDG****Configure Derivative Gain****VALID FROM**  
Version 1.4

| SYNTAX  | UNITS | RANGE          | DEFAULT | ATTRIBUTES           |
|---------|-------|----------------|---------|----------------------|
| <a>CDGn | N/A   | n = 0 - 32,767 | 57      | Immediate, Universal |

**Description** Configure the derivative gain when using PID tuning. This term represents the derivative of position error, in other words the rate at which position error is changing. It produces a damping action in a similar way to velocity feedback and sets the velocity feedback and feedforward gains to equal values.

**\*CFG****Configure Feedforward Gain****VALID FROM**  
Version 1.4

| SYNTAX  | UNITS | RANGE          | DEFAULT | ATTRIBUTES           |
|---------|-------|----------------|---------|----------------------|
| <a>CFGn | N/A   | n = 0 - 32,767 | 57      | Immediate, Universal |

**Description** Used to set the velocity feedforward gain. The opposing action of proportional and velocity feedback results in a position error which depends on speed. This is called 'following error'. Velocity feedforward can be used to offset the following error and improve tracking accuracy. This is important in contouring applications. For true PID, this can be set to the same value as the velocity feedback, or the CDG command may be used instead.

**\*CIG****Configure Integral Gain****VALID FROM**  
Version 1.4

| SYNTAX  | UNITS | RANGE          | DEFAULT | ATTRIBUTES           |
|---------|-------|----------------|---------|----------------------|
| <a>CIGn | N/A   | n = 0 - 32,767 | 0       | Immediate, Universal |

**Description** Used to set the integral gain. If proportional feedback is insufficient to overcome static position errors due to friction, integral action accumulates a steady-state position error until sufficient torque is applied to move the load to reduce the error. It improves overall positioning accuracy, but low frequency oscillation may occur around the commanded position.

|             |                                   |                                  |
|-------------|-----------------------------------|----------------------------------|
| <b>*CIT</b> | <b>Configure In-Position Time</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|-----------------------------------|----------------------------------|

|                       |              |                |                |                          |
|-----------------------|--------------|----------------|----------------|--------------------------|
| <b>SYNTAX</b>         | <b>UNITS</b> | <b>RANGE</b>   | <b>DEFAULT</b> | <b>ATTRIBUTES</b>        |
| <a>CITn               | Milliseconds | n = 0 - 32,767 | 20             | Immediate, Universal     |
| <b>EXECUTION TIME</b> |              |                |                | <b>SEE ALSO</b> CEW, SSC |

**Description**      This command is used to specify the time period that the servo is to be within the in position window before the 'in position' signal is generated. The range of n is 0 to 32,767 and it represents the number of milliseconds to be used as the testing time frame. The shortest time for which the motor must be stopped and within the window before it is considered 'in position' is 2 milliseconds. If at any point during that 2 milliseconds the motor is outside the window, then the 'in position' output will remain inactive.

**Example**

|                |                                |
|----------------|--------------------------------|
| <u>Command</u> | <u>Description</u>             |
| 1SSC1          | Set output 2 as 'in position'  |
| 1CIT30         | Set 'in position' time to 30ms |
| 1CEW20         | Set error window to 20 steps   |

|             |   |                                  |
|-------------|---|----------------------------------|
| <b>*CIW</b> | <b>Configure Integral Action Window</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|---|----------------------------------|

|                       |              |                |                |                      |
|-----------------------|--------------|----------------|----------------|----------------------|
| <b>SYNTAX</b>         | <b>UNITS</b> | <b>RANGE</b>   | <b>DEFAULT</b> | <b>ATTRIBUTES</b>    |
| <a>CIWn               | User steps   | n = 0 - 32,767 | 40             | Immediate, Universal |
| <b>EXECUTION TIME</b> |              |                |                | <b>SEE ALSO</b> OSB  |

**Description**      Sets the width of the of the position window within which integral action is active when the indexer demanded motion is complete.

**Example**

|                |   |
|----------------|---|
| <u>Command</u> | <u>Description</u>  |
| 1CIW5          | Figure 5-6 shows the integral action window set to 5 user steps by this command and it initiates final positioning under integral gain. |

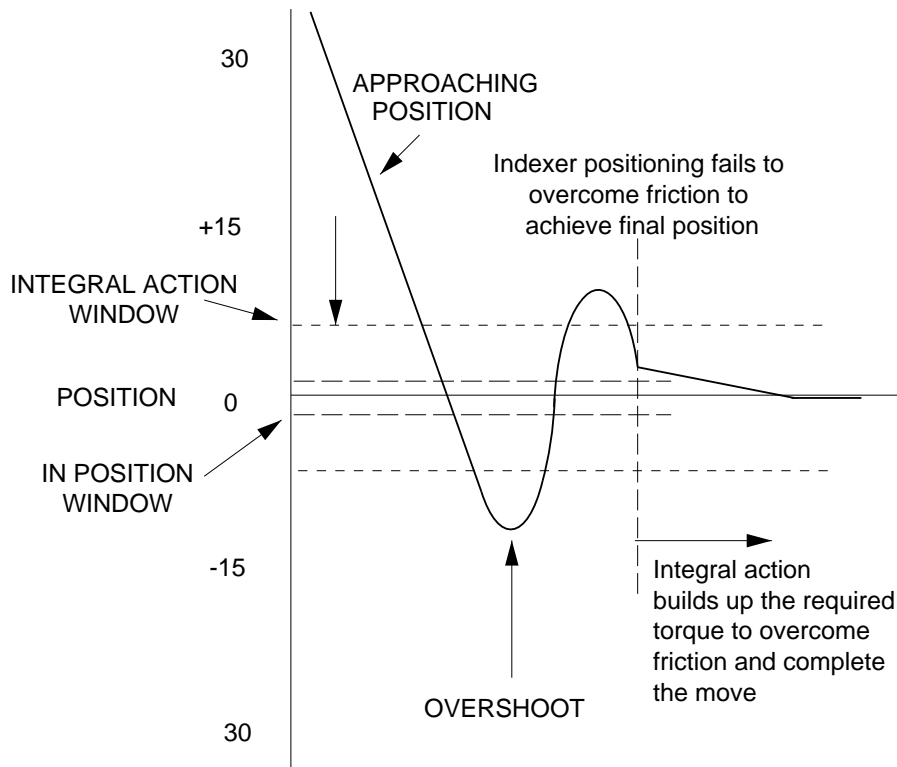


Figure 5-6. Integral Action

|             |  |                                  |
|-------------|--|----------------------------------|
| <b>*CIX</b> | <b>Configure Index Mark Resolution</b> | <b>VALID FROM</b><br>Version 3.4 |
|-------------|--|----------------------------------|

| SYNTAX | UNITS      | RANGE      | DEFAULT | ATTRIBUTES           |
|--------|------------|------------|---------|----------------------|
| <a>CIX | Steps/mark | 1 - 32,767 | 4000    | Immediate, Universal |

**EXECUTION TIME** **SEE ALSO** CUR, CMR, OSK

**Description** For servo systems fitted with an incremental encoder with an index mark, this command programs the RS232C control module with the encoder resolution in terms of the number of steps between index marks. This number is used for encoder error checking (see option OSK). It is usually the number of steps per rev of the motor and it will normally be 2000 or 4000 for a brushless system. This parameter does not need to be changed unless the motor is changed to one having a different encoder resolution or number of steps per revolution.

**Example**

| <u>Command</u>  | <u>Description</u>  |
|-----------------|---|
| <b>1CIX2000</b> | Set the index mark resolution for device 1 for a 500 line encoder |

Note that the control module uses a x4 decoding circuit giving an effective resolution of 4 times the number of encoder lines.

|             |  |                                  |
|-------------|--|----------------------------------|
| <b>*CJL</b> | <b>Enter Combined Motor &amp; Load Inertia</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|--|----------------------------------|

| SYNTAX                | UNITS              | RANGE | DEFAULT                     | ATTRIBUTES                 |
|-----------------------|--------------------|-------|-----------------------------|----------------------------|
| aCJLn                 | Kg-cm <sup>2</sup> | N/A   | 1.6                         | Immediate, Device Specific |
| <b>EXECUTION TIME</b> |                    |       | <b>SEE ALSO</b> HELP11, CTQ |                            |

**Description**      This command has no effect on the behaviour of the RS232C Control Module. It allows entry of the combined motor and load inertia in calculations when using HELP11 as an aid to setting up the servo system parameters.

**Example**

|                                  |  |
|----------------------------------|--|
| <u>Command</u><br><b>1CJL1.8</b> | <u>Description</u><br>Defines a motor and load inertia of 1.8 Kg-cm <sup>2</sup> |
|----------------------------------|--|

|             |                                   |                                  |
|-------------|-----------------------------------|----------------------------------|
| <b>*CMR</b> | <b>Configure Motor Resolution</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|-----------------------------------|----------------------------------|

| SYNTAX                | UNITS     | RANGE          | DEFAULT                  | ATTRIBUTES           |
|-----------------------|-----------|----------------|--------------------------|----------------------|
| <a>CMRn               | Steps/rev | n = 1 - 32,767 | 4000                     | Immediate, Universal |
| <b>EXECUTION TIME</b> |           |                | <b>SEE ALSO</b> CUR, CIX |                      |

**Description**      Programs the RS232C Control Module to match the motor resolution for the servo motor or the open loop stepper motor. The number n is determined by the encoder fitted to the motor or the number of steps of an open loop stepper relative to your required speed and velocity distance units (motor revs, metres, table revs etc.). It will normally be 2000 or 4000 for a brushless system to be programmed in rps units at the motor.

**Example**

|                                   |  |
|-----------------------------------|--|
| <u>Command</u><br><b>1CMR2000</b> | <u>Description</u><br>Set the resolution for device 1 to be programmed in motor revs per second for a 500 line encoder |
|-----------------------------------|--|

Note that the control module uses a x4 decoding circuit giving an effective resolution of four times the number of encoder lines.

|              |                            |                   |
|--------------|----------------------------|-------------------|
| <b>*COFF</b> | <b>Configure Amplifier</b> | <b>VALID FROM</b> |
|--------------|----------------------------|-------------------|

**Offset**

Version 1.4

| SYNTAX   | UNITS | RANGE                | DEFAULT | ATTRIBUTES           |
|----------|-------|----------------------|---------|----------------------|
| <a>COFFn | N/A   | n = +1023 -<br>-1023 | 0       | Immediate, Universal |

**Description** This command can be used to cancel the effect of an offset in the torque amplifier. An excessive offset error can sometimes be the cause of movement at power up, but it is unlikely to affect the closed loop operation unless it is very large. The value should be chosen such that at standstill, position demand equals actual position. It can be set by opening the loop (make CPG, CVG, CIG all zero) and adjusting COFF for zero drift, or reading back the position using the DPE command and setting COFF for the smallest error.

**\*CPE****Configure Position Error**

VALID FROM

Version 1.4

| SYNTAX  | UNITS | RANGE          | DEFAULT | ATTRIBUTES           |
|---------|-------|----------------|---------|----------------------|
| <a>CPEn | Steps | n = 0 - 32,767 | 4000    | Immediate, Universal |

**Description** Defines maximum allowed following error. If the absolute position error is greater than this number, the RS232C Control Module will de-energise the drive. If a valid number in steps is entered, it will become the new maximum following error, otherwise the current setting is reported. Exceeding the maximum following error is a fault condition that will cause the amplifier to be shutdown. If the maximum following error is defined as 32,768 the shutdown function is disabled and no amount of following error will generate an error condition or shutdown the motor.

**NOTE:** In the event that the gain used during setting up is too great, a small position error setting will prevent oscillation and potential mechanical damage to the connected system.

**Example**

Command  
**1CPE1000**

Description  
Set the following error limit of device 1 to 1000 steps

|             |                                    |                                  |
|-------------|------------------------------------|----------------------------------|
| <b>*CPG</b> | <b>Configure Proportional Gain</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|------------------------------------|----------------------------------|

| SYNTAX  | UNITS | RANGE          | DEFAULT | ATTRIBUTES           |
|---------|-------|----------------|---------|----------------------|
| <a>CPGn | N/A   | n = 0 - 32,767 | 92      | Immediate, Universal |

**Description** This command is used to set the proportional gain. The proportional gain determines the amount of torque produced in response to a given position error. It sets the stiffness of the system and also affects the following error. A high proportional gain gives a stiff, responsive system but results in overshoot and oscillation which require damping.

If no value is supplied with the command, the previous setting is reported.

|             |                                       |                                  |
|-------------|---------------------------------------|----------------------------------|
| <b>*CTG</b> | <b>Configure Filter Time Constant</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|---------------------------------------|----------------------------------|

| SYNTAX  | UNITS        | RANGE          | DEFAULT | ATTRIBUTES                 |
|---------|--------------|----------------|---------|----------------------------|
| <a>CTGn | Milliseconds | n = 0 - 32,767 | 3       | Immediate, Device Specific |

**Description** Sets the time constant of the filter used to reduce the effect of transients at the input of the drive amplifier and to smooth the response of the incremental velocity feedback loop.

|             |                           |                                  |
|-------------|---------------------------|----------------------------------|
| <b>*CTQ</b> | <b>Enter Motor Torque</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|---------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                 |
|--------|-------|-------|---------|----------------------------|
| aCTQn  | Nm    | N/A   | 3.6     | Immediate, Device Specific |

**EXECUTION TIME** **SEE ALSO** HELP11, CJL

**Description** This command has no effect on the behaviour of the RS232C Control Module. It allows entry of the available motor torque in calculations when using HELP11 as an aid to setting up the servo system parameters.

**Example**

|                                  |   |
|----------------------------------|---|
| <u>Command</u><br><b>2CTQ2.4</b> | <u>Description</u><br>Defines the maximum motor torque as 2.4Nm |
|----------------------------------|---|

**\*CUR****Configure User Resolution****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;CURn

**UNITS**

User steps

**RANGE**

n = 1 - 32,767

**DEFAULT**

4000

**ATTRIBUTES**

Immediate, Universal

**EXECUTION TIME****SEE ALSO** CMR**Description**

This command is used to define the number of steps per revolution required by the user. Previous D commands are reconfigured by the CUR command so that the move distance remains the same.

**Example 1**Command

**1CUR2000**  
**D4000**  
**G**

Description

Set user resolution as 2000 steps/rev  
 Set distance to 4000 user steps (2 revs)  
 Go - move 2 revolutions

**Example 2**Command

**1CUR4000**  
**1D4000**  
**1CUR2000**

Description

Set user resolution to 4000 steps/rev  
 Set the move distance to 4000 steps (1 rev)  
 Change the resolution to 2000 step/rev, but the distance programmed by the D command changes to 2000 so the move distance remains 1 revolution.

**\*CVG****Configure Velocity Gain****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;CVGn

**UNITS**

N/A

**RANGE**

n = 0 - 32,767

**DEFAULT**

57

**ATTRIBUTES**

Immediate, Universal

**Description**

Used to set the gain of the velocity feedback loop. The velocity feedback is a signal which increases with shaft speed. It acts in opposition to proportional feedback to stabilise the motion. This setting is generally used to damp vibrations in the servo response, allowing a higher proportional gain to be used. See 'Setting Up Servo Parameters' in Section H for further details

|          |                 |                                  |
|----------|-----------------|----------------------------------|
| <b>D</b> | <b>Distance</b> | <b>VALID FROM</b><br>Version 1.4 |
|----------|-----------------|----------------------------------|

| SYNTAX   | UNITS | RANGE                           | DEFAULT | ATTRIBUTES                                  |
|----------|-------|---------------------------------|---------|---|
| <a>D<s>n | Steps | -268,435,455 to<br>+268,435,455 | 0       | Buffered, Universal,<br>Savable in Sequence |

|                       |   |
|-----------------------|---|
| <b>EXECUTION TIME</b> | <b>SEE ALSO</b> MN, MPA,<br>MPI, A, V,<br>G, MQ |
|-----------------------|---|

**Description** This command is used in MN and MQ to move the motor a number of steps (n ) in the direction specified by <s>. The direction is assumed to be positive if no sign is given. The D command overrides a previous H command in terms of motor direction.

**Example**

| <u>Command</u>  | <u>Description</u>  |
|-----------------|---|
| <b>MN</b>       | Normal mode   |
| <b>A10</b>      | Set acceleration to 10 revs/sec <sup>2</sup>                              |
| <b>V10</b>      | Set velocity to 10 revs/sec   |
| <b>D-100000</b> | A 4000 steps/rev motor will turn 25 revolutions in the negative direction |
| <b>G</b>        | Go  |

|            |                              |                                  |
|------------|------------------------------|----------------------------------|
| <b>DFX</b> | <b>Display Flags Indexer</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|------------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                                 |
|--------|-------|-------|---------|--|
| aDFX   | N/A   | N/A   | N/A     | Immediate, Device Specific,<br>Never Saved |

**RESPONSE TO aDFX IS** \*bbbb\_.....

**Description** Reports all indexer status flags as 32 bits where the response is:

bbbb\_.....bbb[CR]

The order of the bits is \*31,30,29,28\_.....\_3,2,1,0\*

Response Bits:-

31-27 are all reserved for future use

- 26 1= Paused by PSY command
- 25 1= Paused waiting for SYNC input
- 24 1= Paused waiting for registration trigger
- 23 1= Run sequence on power up
- 22 1= Executing a sequence
- 21 1= Paused, waiting on in-position
- 20 1= Paused, waiting on distance trigger bits
- 19 1= Paused, waiting on trigger bits
- 18 1= Paused by U command flag, waiting on a C (continue)
- 17 1= Paused by PS command, waiting on C (continue)
- 16 1= Performing a wait
- 15 1= Homing 2nd leg, low speed move back to home limit
- 14 1= Homing 1st leg of home move, high speed to home limit
- 13 1= Go home move to encoder position
- 12 1= Home limit switch has been hit
- 11 1= +limit switch has been hit
- 10 1= +limit switch has been hit
- 9 1= Jogging is enabled, we are jogging
- 8 1= Skip next buffered command
- 7 1= Set if continuous move direction is negative
- 6 1= Set if current move direction is negative
- 5 1= Set if we want to change velocity
- 4 1= Set if in continuous mode, clear if in preset mode
- 3 1= Set if in absolute mode, clear if in incremental mode
- 2 1= Performing a variable speed move
- 1 1= Performing a preset move
- 0 1= Performing a continuous move

| <b>DIC</b>         | <b>Display Indexer Counter</b>  | <b>VALID FROM</b><br>Version 1.4   |
|--------------------|---|--|
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>   |
| aDIC               | User steps  | -2,097,152 to +2,097,151   |
|                    |   | <b>DEFAULT</b>   |
|                    |   | N/A  |
|                    |   | <b>ATTRIBUTES</b>  |
|                    |   | Immediate, Device Specific, Never Saved  |
|                    | <b>EXECUTION TIME</b>   | <b>SEE ALSO</b> D  |
|                    | <b>RESPONSE TO</b> aDPa IS *n   |  |
| <b>Description</b> | Requests a single display of the contents of the indexer counter as a single value in steps at the user resolution. This is the programmed D value. |  |
| <b>Example</b>     | <u>Command</u><br><b>1DIC</b>   | <u>Response</u><br>*1000 (CR). The counter is programmed for a D value of 1000 user steps. |

| <b>DPA</b>         |   | <b>Display Position Actual</b> |                | <b>VALID FROM</b><br>Version 1.4           |
|--------------------|---|--------------------------------|----------------|--|
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>                   | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                          |
| aDPA               | User steps  | -32,767 to<br>+32,768          | N/A            | Immediate, Device Specific,<br>Never Saved |
|                    |   | <b>EXECUTION TIME</b>          |                | <b>SEE ALSO</b> OSC, PZ,<br>SP             |
|                    |   | <b>RESPONSE TO</b> aDPA IS *n  |                |  |
| <b>Description</b> | Continuous display of actual position. The response is the position in user steps which should have resulted from the number of clock pulses sent to the drive from the RS232C Control Module since the drive was enabled or a PZ or SP command was issued, provided that the motor did not de-synchronise. |                                |                |  |

| <b>DPE</b>         |   | <b>Display Position Error</b> |                | <b>VALID FROM</b><br>Version 1.4           |
|--------------------|---|-------------------------------|----------------|--|
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>                  | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                          |
| aDPE               | User steps  | -32,767 to<br>+32,768         | N/A            | Immediate, Device Specific,<br>Never Saved |
| <b>Description</b> | Continuous display of position error. The response is the difference between the setpoint and the actual position in user steps. It is used by the position control algorithm to control motor current. The difference between command setpoint and actual position is also used to determine if the motor is within the deadband specified by the CEW command. The response is a single instantaneous value reported at 150ms intervals until the return key is pressed. See also OSC command. |                               |                |  |

| <b>DPS</b>         |  | <b>Display Position Setpoint</b> |                | <b>VALID FROM</b><br>Version 1.4           |
|--------------------|--|----------------------------------|----------------|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                     | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                          |
| **aDPS             | User steps   | +268,435,455 to<br>-268,435,455  | N/A            | Immediate, Device Specific,<br>Never Saved |
| <b>Description</b> | Continuous display of the commanded position. The response is repeatedly updated until the return key is pressed. It is the absolute number of pulses sent to the drive from the interface since the drive was enabled (or reset). See also OSC command. |                                  |                |  |

|           |                       |                                  |
|-----------|-----------------------|----------------------------------|
| <b>DR</b> | <b>Display Report</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-----------------------|----------------------------------|

| SYNTAX | UNITS | RANGE  | DEFAULT | ATTRIBUTES                              |
|--------|-------|--------|---------|---|
| aDR<n> | N/A   | 1 or 2 | N/A     | Immediate, Device Specific, Never Saved |

**EXECUTION TIME** **SEE ALSO** PS,U

**Description** This command reports the setup of the various parameters of the control module:

If n = 1, the general status of the control module parameters is displayed.

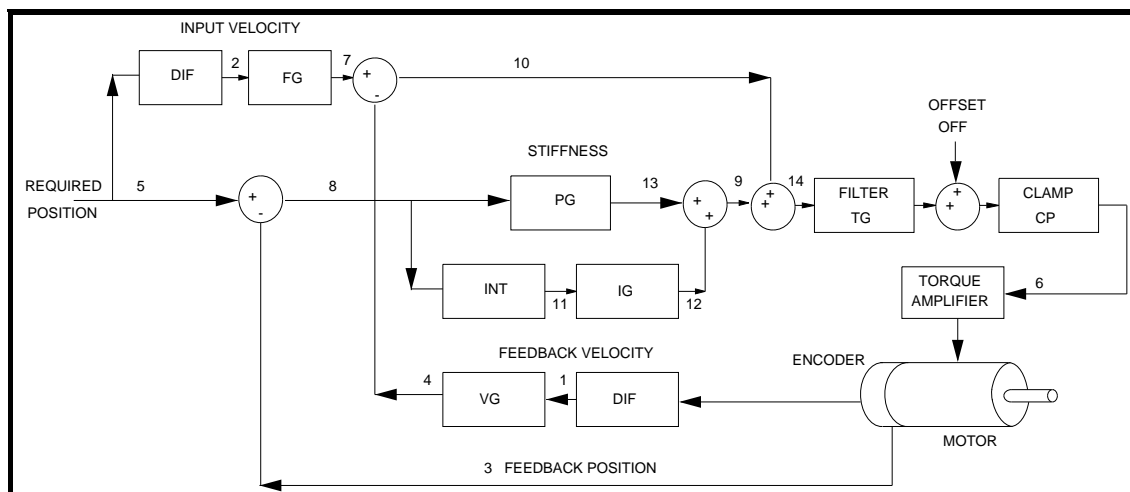
If n = 2, the status of the selectable facilities is displayed.

If n is omitted, both the control module parameters and the selectable facilities are shown on consecutive displays.

|           |                       |                                  |
|-----------|-----------------------|----------------------------------|
| <b>DS</b> | <b>Display Signal</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-----------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                              |
|--------|-------|-------|---------|---|
| aDSn   | N/A   | N/A   | N/A     | Immediate, Device Specific, Never Saved |

**Description** Requests the continuous display of variable parameters. The PIC screen shown below is a representation of the control algorithm and any parameter in this diagram can be displayed. If n is omitted, a list of the signal displays is produced. See also OSC command.



**Figure 5-7. Servo Control Loop**

The individual parameters are:-

|     |                        |      |                               |
|-----|------------------------|------|-------------------------------|
| DS1 | Motor velocity         | DS8  | Position error                |
| DS2 | Input velocity         | DS9  | Demand less filtered velocity |
| DS3 | Motor position         | DS10 | Velocity action difference    |
| DS4 | Feedback velocity      | DS11 | Integral of position error    |
| DS5 | Input position         | DS12 | Integral action               |
| DS6 | Filtered torque demand | DS13 | Error times gain (= torque)   |
| DS7 | Input velocity action  | DS14 | Torque demand                 |

**Example**

Command

**1DS8**

Description

Shows the position error for RS232C Control Module 1

The significance of the numbers displayed will vary with the signal point chosen. Input and motor velocities are expressed in encoder edges per 500µS period, with a maximum value of +/-255. All other parameters have values between +32767 and -32768, except the torque demand which covers the range +1023 to -1024 for full torque. A continuous display of the maximum value indicates saturation.

|             |                             |  |
|-------------|-----------------------------|--|
| <b>*DTA</b> | <b>Set Dither Amplitude</b> | <b>Not used in System 7,<br/>PS7X, BLX</b> |
|-------------|-----------------------------|--|

|             |                             |  |
|-------------|-----------------------------|--|
| <b>*DTF</b> | <b>Set Dither Frequency</b> | <b>Not used in System 7,<br/>PS7X, BLX</b> |
|-------------|-----------------------------|--|

| <b>DVA</b>         |  | <b>Display Velocity Actual</b> |                | <b>VALID FROM</b>                       |
|--------------------|--|--------------------------------|----------------|---|
|                    |  |                                |                | Version 1.4                             |
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                   | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                       |
| aDVA               | Steps/500 $\mu$ s  | $\pm$ 255                      | N/A            | Immediate, Device Specific, Never Saved |
| <b>Description</b> | This command returns a continuous display of the actual velocity. The number is reported in motor steps per 500 $\mu$ s and is repeatedly updated until the return key is pressed. This value is the shaft velocity being read from the encoder measured over a 500 $\mu$ s period. The DVA command has the same effect as the DS1 command. See also OSC command |                                |                |   |

| <b>DVS</b>         |   | <b>Display Velocity Setpoint</b> |                | <b>VALID FROM</b>                       |
|--------------------|---|----------------------------------|----------------|---|
|                    |   |                                  |                | Version 1.4                             |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>                     | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                       |
| aDVS               | Steps/500 $\mu$ s   | $\pm$ 255                        | N/A            | Immediate, Device Specific, Never Saved |
| <b>Description</b> | Requests a continuous display of the velocity setpoint. The displayed value is the velocity being sent to the velocity part of the servo loop by the servo algorithm. It is repeatedly updated until the return key is pressed. The DVS command has the same effect as the DS2 command. See also OSC command. |                                  |                |   |

| <b>E</b>           |   | <b>Enable Communications</b> |                | <b>VALID FROM</b>    |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
|--------------------|---|------------------------------|----------------|----------------------|----------------|--------------------|----------|---------------------------------|-----------|------------------------------|-----------|------------------------------|------------|--|-----------|----------------------------|--------------|----------------------------|----------|------------------------|
|                    |   |                              |                | Version 1.4          |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>                 | <b>DEFAULT</b> | <b>ATTRIBUTES</b>    |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <a>E               | N/A   | N/A                          | N/A            | Immediate, Universal |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>Description</b> | This command allows the RS232C Control Module to accept commands over the serial communications interface. You can re-enable the the communications interface with this command if you had previously disabled the RS232C interface with the F command. If several units are using the same communications interface, use of the E and F commands can help to streamline programming.   |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>Example</b>     | <table> <thead> <tr> <th><u>Command</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td><b>F</b></td> <td>Disable communications all axes</td> </tr> <tr> <td><b>1E</b></td> <td>Enable communications axis 1</td> </tr> <tr> <td><b>4E</b></td> <td>Enable communications axis 4</td> </tr> <tr> <td><b>A10</b></td> <td>Set acceleration to 10 revs/sec<sup>2</sup></td> </tr> <tr> <td><b>V5</b></td> <td>Set velocity to 5 revs/sec</td> </tr> <tr> <td><b>D5000</b></td> <td>Set distance to 5000 steps</td> </tr> <tr> <td><b>G</b></td> <td>Only axes 1 and 4 move</td> </tr> </tbody> </table> |                              |                |                      | <u>Command</u> | <u>Description</u> | <b>F</b> | Disable communications all axes | <b>1E</b> | Enable communications axis 1 | <b>4E</b> | Enable communications axis 4 | <b>A10</b> | Set acceleration to 10 revs/sec <sup>2</sup> | <b>V5</b> | Set velocity to 5 revs/sec | <b>D5000</b> | Set distance to 5000 steps | <b>G</b> | Only axes 1 and 4 move |
| <u>Command</u>     | <u>Description</u>  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>F</b>           | Disable communications all axes   |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>1E</b>          | Enable communications axis 1  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>4E</b>          | Enable communications axis 4  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>A10</b>         | Set acceleration to 10 revs/sec <sup>2</sup>  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>V5</b>          | Set velocity to 5 revs/sec  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>D5000</b>       | Set distance to 5000 steps  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |
| <b>G</b>           | Only axes 1 and 4 move  |                              |                |                      |                |                    |          |                                 |           |                              |           |                              |            |  |           |                            |              |                            |          |                        |

| <b>EX</b>          |  | <b>Set Communication Style</b> |                          | <b>VALID FROM</b>           |
|--------------------|--|--------------------------------|--------------------------|-----------------------------|
|                    |  |                                |                          | Version 1.4                 |
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                   | <b>DEFAULT</b>           | <b>ATTRIBUTES</b>           |
| aEXn               | N/A  | n = 0 or 1                     | User messages sent (EX1) | Immediate, Device Specific, |
| <b>Description</b> | Sets the style of communication between the RS232C Control Module and the terminal/computer. |                                |                          |                             |
|                    | n = 1 sets terminal mode i.e. user-friendly messages.  |                                |                          |                             |
|                    | n = 0 sets computer mode i.e. no user-friendly messages are sent.                            |                                |                          |                             |
|                    | The ">" prompt is not returned when n = 0  |                                |                          |                             |

**F****Disable  
Communications****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;F

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**

Immediate, Universal,

**Description**

The F command disables command execution by devices connected to the RS232C Control Module. The disabled device will still echo back commands sent to it however. It is useful when you are programming multiple devices on a single interface. Devices that are not intended to respond to universal commands should be disabled using device specific F commands. This enables you to program other devices without specifying a device identifier on every command.

**Example**Command

1F

3F

G

Description

Disable axis 1

Disable axis 3

All axes except 1 &amp; 3 will move

**FDA****Acceleration, Linear  
Interpolation Moves****Not used in PS7X, BLX**

|            |   |                              |
|------------|---|------------------------------|
| <b>FDV</b> | <b>Velocity, Linear Interpolation Moves</b> | <b>Not used in PS7X, BLX</b> |
|------------|---|------------------------------|

|            |                          |                                  |
|------------|--------------------------|----------------------------------|
| <b>FOL</b> | <b>Following Percent</b> | <b>VALID FROM</b><br>Version 3.2 |
|------------|--------------------------|----------------------------------|

|                 |                  |                          |                |                                    |
|-----------------|------------------|--------------------------|----------------|------------------------------------|
| SYNTAX<br>aFOLn | UNITS<br>Percent | RANGE<br>0.0 -<br>5000.0 | DEFAULT<br>100 | ATTRIBUTES<br>Buffered, Universal, |
|-----------------|------------------|--------------------------|----------------|------------------------------------|

**EXECUTION TIME** **SEE ALSO** SIM, CCS

**Description** During preset following indexing mode (SIM5) the move velocity in rps is set as a percentage of the following input velocity in rps by the FOL command value.

**Example**

| <u>Command</u> | <u>Description</u>  |
|----------------|---|
| <b>CUR800</b>  | Pulse source resolution in pulses / rev   |
| <b>1CCS3</b>   | With clock and direction decode   |
| <b>CMR4000</b> | The interface motor resolution  |
| <b>MN</b>      | Start linear interpolation move   |
| <b>MPI</b>     | D values are incremental  |
| <b>1SIM5</b>   | Set indexer mode to preset following  |
| <b>FOL50.0</b> | Follow at 50% of the input encoder rate   |
| <b>A100</b>    | Acceleration fixed at 100 rps/s   |
| <b>D16000</b>  | Will move 4 revs  |
| <b>G</b>       | The motor will accelerate at 100 rps absolute to 1/2 the input encoder speed in rps |

The CMR / CUR ratio matches the input/output pulse rate, allowing FOL to control the output velocity in rps.

| <b>G</b>              |   | <b>Go</b>    |                                     | <b>VALID FROM</b>                           |
|-----------------------|---|--------------|-------------------------------------|---|
|                       |   |              |                                     | Version 1.4                                 |
| <b>SYNTAX</b>         | <b>UNITS</b>  | <b>RANGE</b> | <b>DEFAULT</b>                      | <b>ATTRIBUTES</b>                           |
| <a>G                  | N/A   | N/A          | N/A                                 | Buffered, Universal,<br>Savable in Sequence |
| <b>EXECUTION TIME</b> |   |              |                                     | <b>SEE ALSO</b> S, MN, MC,<br>MQ, A, V, D   |
| <b>Description</b>    | Go - make a move using the previously entered parameters. It is not necessary to re-enter A, V and D. |              |                                     |   |
| <b>Example</b>        | <u>Command</u>  |              | <u>Description</u>                  |   |
|                       | MN  |              | Select normal mode                  |   |
|                       | A10   |              | Set acceleration to 10 revs/sec/sec |   |
|                       | V10   |              | Set velocity to 10 revs/sec         |   |
|                       | D100000   |              | Set distance to 100,000 steps       |   |
|                       | G   |              | Go                                  |   |

| <b>GA</b>             |  | <b>Go Home Acceleration</b> |                | <b>VALID FROM</b>                           |
|-----------------------|--|-----------------------------|----------------|---|
|                       |  |                             |                | Version 1.4                                 |
| <b>SYNTAX</b>         | <b>UNITS</b>   | <b>RANGE</b>                | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>GAn                | Revs/sec <sup>2</sup>  | n = 0.001 -<br>999,999      | 10             | Buffered, Universal,<br>Savable in Sequence |
| <b>EXECUTION TIME</b> |  |                             |                | <b>SEE ALSO</b> GH, SS,<br>GHF, GHP         |
| <b>Description</b>    | This command is used to set the acceleration rate to be used in performing the GH command. The value can be saved in non-volatile RAM. The default is changed by the user if the GA command is issued and then a save is performed (SV command). |                             |                |   |

| <b>GH</b>             |  | <b>Go Home</b>      |                | <b>VALID FROM</b>                           |
|-----------------------|--|---------------------|----------------|---|
|                       |  |                     |                | Version 1.4                                 |
| <b>SYNTAX</b>         | <b>UNITS</b>   | <b>RANGE</b>        | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>GH<s>n             | Revs/sec   | n = 0.0001 -<br>100 | N/A            | Buffered, Universal,<br>Savable in Sequence |
| <b>EXECUTION TIME</b> |  |                     |                | <b>SEE ALSO</b> RG                          |
| <b>Description</b>    | Go Home + or -. This command causes the RS232C Control Module to rotate the motor in the direction and at the speed specified until its home limit input is activated. |                     |                |   |

The GH+2 command causes the controller to seek the home position at 2 revs/sec. The sign is optional (a "+" or positive is assumed if omitted). The controller will reverse direction if a limit is activated and it will cease the attempt to go home if the second limit is also activated.

Note that the GH command will reset the absolute position counter to zero. If you want to set the counter to another value, use the SP command after the GH is complete.

**Example**

Command  
**GH-2**

Description  
Motor will turn negative at 2 revolutions per second and look for Home limit input to go active. It will then stop, reverse and finally stop completely when the home switch is hit for the second time.

|                    |  |                        |                |   |
|--------------------|--|------------------------|----------------|---|
| <b>GHF</b>         |  | <b>Go Home Final</b>   |                | <b>VALID FROM</b><br>Version 1.4            |
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>           | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>GHFn            | Revs/sec   | n = 0.01 - 100         | 0.1            | Buffered, Universal,<br>Savable in Sequence |
|                    |  | <b>EXECUTION TIME</b>  |                | <b>SEE ALSO</b> GH, SS, GA                  |
| <b>Description</b> | This command is used to set the velocity for the final move in the go home sequence. |                        |                |   |
|                    | The value can be saved in battery backed up RAM.                                     |                        |                |   |
| <b>GL</b>          |  | <b>Go, Linear Mode</b> |                | <b>Not used in PS7X, BLX</b>                |

**GLD****Go Linear Define**

Not used in PS7X, BLX

**GSY****Go, Synchronous Mode**

Not used in PS7X, BLX

**H****Change Direction****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;H

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**Buffered, Universal,  
Savable in Sequence**EXECUTION TIME****SEE ALSO** H+, H-, D**Description**

This command reverses direction of the next move.  
The D command could subsequently be used to reset the direction.

**Example**Command**D8000****G****H****G**Description

Set distance to 8000 steps

Go - move 8000 steps in the + direction

Reverse direction

Go - move 8000 steps in the - direction

**^H****Backspace****VALID FROM**

Version 1.4

**SYNTAX**

^H

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**

Immediate, Never Saved

**Description**

This produces a backspace during command input, deleting the last

character. It will not prevent the execution of an immediate command.

The ^H command (^H indicates that the CONTROL or CTRL key is held down when the H key is pressed) backspaces one character provided a delimiter has not been sent. A new character may be entered at that position to replace the existing character. The effect of this command character is to cause the RS232C Control Module to back up one character in the command buffer regardless of what appears on the terminal. On some terminals pressing the BACKSPACE key will produce the same character.

Because the RS232C Control Module processes each command upon receipt of the delimiter, it is not possible to backspace once the delimiter is encountered. If you type a device address wrongly, you must re-type the whole command.

|                       |   |                      |   |  |
|-----------------------|---|----------------------|---|--|
| <b>H+ &amp; H-</b>    |   | <b>Set Direction</b> |   | <b>VALID FROM</b><br>Version 1.4         |
| <b>SYNTAX</b>         | <b>UNITS</b>  | <b>RANGE</b>         | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                        |
| <a>Hs                 | N/A   | N/A                  | N/A   | Buffered, Universal, Savable in Sequence |
| <b>EXECUTION TIME</b> |   |                      | <b>SEE ALSO</b> H, D  |  |
| <b>Description</b>    | Sets the direction of all moves according to s = "+" (positive) or "-" (negative). A subsequent D command will reset the direction for following moves. |                      |   |  |
| <b>Example</b>        | <u>Command</u><br>H-  |                      | <u>Description</u><br>All moves are made negative until otherwise specified in the command string |  |

|                    |  |                             |                |                                   |
|--------------------|--|-----------------------------|----------------|-----------------------------------|
| <b>HELP</b>        |  | <b>Produce Help Screens</b> |                | <b>VALID FROM</b><br>Version 1.4  |
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                 |
| <a>HELP<n>         | N/A  | n = 1 - 13                  | N/A            | Immediate, Universal, Never Saved |
| <b>Description</b> | Displays the help screens. It provides a list of commands and a brief description of each command. The command on its own will produce the base menu followed by all other help screens. Individual screens can be accessed by typing HELP followed by the appropriate screen number. HELP 11 provides a worked example on servo set up. |                             |                |                                   |

This is available in terminal mode only (see EX command).

The CR key will take you out of the help menu.

With the exception of HELP 11, all interfaces return the same help information, so no device address is needed. The gain values returned in HELP 11 relate to the torque and inertia figures programmed by the CJL and CTQ commands.

|           |   |                                  |
|-----------|---|----------------------------------|
| <b>HS</b> | <b>Display Hardware Configured Switches</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|---|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                              |
|--------|-------|-------|---------|---|
| aHS    | N/A   | N/A   | N/A     | Immediate, Device Specific, Never Saved |

**Description** Display hardware configured switches i.e. the way in which the interface has been configured for its environment. The command displays a binary representation of this status. The response is \*bbbb\_bbbb where the order of the bits is:

ABCD\_XXXX

Therefore bit 1 represents the status of HSA, bit 2 the status of HSB etc. Bits 5-8 are not used.

|                                    |                                   |
|------------------------------------|-----------------------------------|
| HSA: 1 = Resolver                  | 0 = Other (not available)         |
| HSB: 1 = Brush                     | 0 = Brushless initialisation      |
| HSC: 1 = Servo                     | 0 = Open loop                     |
| HSD: 0 = No daughter board         | 1 = Output expansion board fitted |
| 2 = Analogue feedback board fitted |                                   |

**Example**

Command  
**1HS**

Response  
\*0010\_0000 This indicates that the interface is configured for brushless servo operation

**HSB****Configure Motor Type**
**VALID FROM**  
Version 1.4
**SYNTAX**

aHSBn

**UNITS**

N/A

**RANGE**

n = 0 , 1

**DEFAULT**

0

**ATTRIBUTES**Immediate, Device Specific,  
Never Saved**EXECUTION TIME****SEE ALSO** HSC**Description**

Sets the configuration for drive commutation;

n = 0: non self-commutating drive (System 7)

n = 1: self-commutating drive (BL &amp; BR)

|                                   | <b>HSB</b> | <b>HSC</b> |
|-----------------------------------|------------|------------|
| <b>Stepper</b>                    | 1          | 0          |
| <b>Hybrid Servo</b>               | 0          | 1          |
| <b>Brushless Servo (System 7)</b> | 0          | 1          |
| <b>Brushless Servo (BL)</b>       | 1          | 1          |
| <b>DC Brush Servo</b>             | 1          | 1          |

HSB &amp; HSC Combined Commands

**HSC****Configure Motor Type**
**VALID FROM**  
Version 2.0
**SYNTAX**

aHSCn

**UNITS**

N/A

**RANGE**

n = 1, 0

**DEFAULT**

0

**ATTRIBUTES**Immediate, Device Specific,  
Never Saved**EXECUTION TIME****SEE ALSO** HSB**Description**

Sets the configuration for a stepper motor;

n = 0: open loop stepper

n = 1: other

|                                   | <b>HSB</b> | <b>HSC</b> |
|-----------------------------------|------------|------------|
| <b>Stepper</b>                    | 1          | 0          |
| <b>Hybrid Servo</b>               | 0          | 1          |
| <b>Brushless Servo (System 7)</b> | 0          | 1          |
| <b>Brushless Servo (BL)</b>       | 1          | 1          |
| <b>DC Brush Servo</b>             | 1          | 1          |

HSB &amp; HSC Combined Commands

|            |                                   |                        |
|------------|-----------------------------------|------------------------|
| <b>HSD</b> | <b>Daughter Board<br/>Outputs</b> | <b>NOT USED IN BLX</b> |
|------------|-----------------------------------|------------------------|

|           |                           |                                  |
|-----------|---------------------------|----------------------------------|
| <b>ID</b> | <b>Immediate Distance</b> | <b>VALID FROM</b><br>Version 3.1 |
|-----------|---------------------------|----------------------------------|

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b>                    | <b>DEFAULT</b> | <b>ATTRIBUTES</b>     |
|---------------|--------------|---------------------------------|----------------|-----------------------|
| <a>ID<s>n     | Steps        | -268,435,455<br>to +268,435,455 | 4000           | Immediate, Universal, |

**SEE ALSO** D

**Description** During motion, the distance the motor is travelling can be altered to a new value set by the ID command. An attempt to change the motion direction or to set the target within the current stopping distance causes the unit to stop immediately at the currently programmed acceleration rate - if you are in terminal mode a warning message is returned. In incremental mode the sign is ignored and the distance is relative to the original start point of the current move, and in the same direction of motion. When stopped the ID command is simply an immediate version of the D command.

**Example**

| <u>Command</u>  | <u>Description</u>            |
|-----------------|-------------------------------|
| <b>MN</b>       | Normal preset movement        |
| <b>MPA</b>      | Absolute programming          |
| <b>PZ</b>       | Force current position 0      |
| <b>D-140000</b> | Target is -140000             |
| <b>G</b>        | Start moving                  |
| <b>ID-40000</b> | Change target                 |
| <b>IPR</b>      | Reports -40000 at end of move |

|           |                         |                   |
|-----------|-------------------------|-------------------|
| <b>IO</b> | <b>Immediate Output</b> | <b>VALID FROM</b> |
|-----------|-------------------------|-------------------|

Version 1.4

| SYNTAX   | UNITS | RANGE       | DEFAULT | ATTRIBUTES                        |
|----------|-------|-------------|---------|-----------------------------------|
| <a>IObbb | N/A   | b = 0, 1, X | N/A     | Immediate, Universal, Never Saved |

**Description** This command sets the output bits as specified in the pattern on an immediate basis. b = 0, 1 or X (X leaves the output unchanged). If the optional output daughter board is fitted (HSD=1), the syntax becomes <a>IObbbbbb.

**Example**

Command  
**2IO10X**

Description  
This command sets the outputs of device 2 : O1 to 1 and O2 to 0. It leaves O3 unchanged.

**IS****Input Status**

**VALID FROM**  
Version 1.4

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                              |
|--------|-------|-------|---------|---|
| aIS    | N/A   | N/A   | N/A     | Immediate, Device Specific, Never Saved |

**Description** Report Input status. The response is:

\*bbbbbbb\_bbbbbbb[CR]

The pattern of the response is, from left to right:

HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX TRACK \_ INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

This is an immediate command that will report the status of all the inputs no matter what the configuration.

Open circuit inputs are indicated by 0's.

**Example**

Command  
**1IS**

Response  
\*0000000\_1000000[CR]. This indicates that only Input 1 is active

**IV****Immediate Velocity****VALID FROM**  
Version 3.1

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b>            | <b>DEFAULT</b> | <b>ATTRIBUTES</b>    |
|---------------|--------------|-------------------------|----------------|----------------------|
| aIVn          | Revs/sec     | n = 0.0001 to<br>100.00 | 1              | Immediate, Universal |

**SEE ALSO** V

**Description** This is an immediate version of the V command. During motion, the speed the motor is travelling can be altered to a new value set by the IV command. In response to the IV command the motor will accelerate or decelerate to the new velocity, but will still travel the programmed distance.

The IV command uses the RS232C communication link to alter the motor velocity. If the motor is already stopping at the end of a move a warning message is returned (terminal mode only).

**Example**

| <u>Command</u>  | <u>Response</u>              |
|-----------------|------------------------------|
| <b>MN</b>       | Normal preset movement       |
| <b>MPA</b>      | Absolute programming         |
| <b>V5</b>       |                              |
| <b>D-140000</b> | Target is -140000            |
| <b>G</b>        | Start moving                 |
| <b>IV20</b>     | Speed up during move         |
| <b>1PR</b>      | Shows end point is unchanged |

**JA****Jog Acceleration****VALID FROM**  
Version 1.4

| <b>SYNTAX</b> | <b>UNITS</b>          | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
|---------------|-----------------------|-----------------------|----------------|---|
| <a>JAn        | Revs/sec <sup>2</sup> | n = 0.01 -<br>999,999 | 99             | Buffered, Universal,<br>Savable in Sequence |

**EXECUTION TIME****SEE ALSO** JV

**Description** Sets the jog acceleration. The acceleration rate used in jog operations is set by this command.

**Example**

| <u>Command</u> | <u>Description</u>                      |
|----------------|---|
| <b>JV1.5</b>   | Set jog speed to 1.5 revs/sec           |
| <b>JA20</b>    | Set jog acceleration to 20 revs/sec/sec |

|           |                     |                                  |
|-----------|---------------------|----------------------------------|
| <b>JV</b> | <b>Jog Velocity</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|---------------------|----------------------------------|

| SYNTAX | UNITS    | RANGE            | DEFAULT | ATTRIBUTES                               |
|--------|----------|------------------|---------|--|
| <a>JVn | Revs/sec | n = 0.0001 - 100 | 1       | Buffered, Universal, Savable in Sequence |

**EXECUTION TIME** **SEE ALSO** JA

**Description** Sets the jog velocity. The velocity used in jog operations is set using this command. The factory default can be changed by saving a new JV value to non-volatile RAM using the SV command.

|          |             |                                  |
|----------|-------------|----------------------------------|
| <b>K</b> | <b>Kill</b> | <b>VALID FROM</b><br>Version 1.4 |
|----------|-------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                        |
|--------|-------|-------|---------|-----------------------------------|
| <a>K   | N/A   | N/A   | N/A     | Immediate, Universal, Never Saved |

**EXECUTION TIME** **SEE ALSO** S, LS

**Description** Kill - This command stops RS232C Control Module commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

|             |             |                                  |
|-------------|-------------|----------------------------------|
| <b>KILL</b> | <b>Kill</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|-------------|----------------------------------|

| SYNTAX  | UNITS | RANGE | DEFAULT | ATTRIBUTES                        |
|---------|-------|-------|---------|-----------------------------------|
| <a>KILL | N/A   | N/A   | N/A     | Immediate, Universal, Never Saved |

**EXECUTION TIME** **SEE ALSO** S, LS

**Description** This is an alternative expression for the K command. It stops RS232C Control Module commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a command sequence download in progress (XD command). The command buffer is also cleared.

|          |             |                                  |
|----------|-------------|----------------------------------|
| <b>L</b> | <b>Loop</b> | <b>VALID FROM</b><br>Version 1.4 |
|----------|-------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES |
|--------|-------|-------|---------|------------|
|--------|-------|-------|---------|------------|

<a>Ln            Times            n = 0 - 65545    N/A            Buffered, Universal, Savable in Sequence

**EXECUTION TIME**

**SEE ALSO**    Y, N

**Description**

When combined with the N command, the L command will cause all of the commands between L and N to be repeated the number of times indicated by n. If L is entered with no number following it or if n = 0, the commands will be repeated continuously.

The END-OF-LOOP command (N) can be used to indicate that the RS232C Control Module should proceed with further commands after the designated numbers of loops have been executed, or in combination with the "Y" command, to indicate where execution is to stop. The "U" command may be used to temporarily halt loop execution, the C command will then cause the loop to resume execution.

There should be a balanced number of loops and loop terminators inside a sequence. Starting a loop in one sequence and terminating it in another sequence is not allowed. Nested loops require complete closure before execution will begin.

**Example**

| <u>Command</u> | <u>Description</u>                      |
|----------------|---|
| <b>PS</b>      | Pause                                   |
| <b>A10</b>     | Set acceleration to 10 rps <sup>2</sup> |
| <b>V10</b>     | Set velocity to 10 revs/sec             |
| <b>D1000</b>   | Set distance to 1,000 steps             |
| <b>L5</b>      | Loop 5 times                            |
| <b>G</b>       | Go                                      |
| <b>N</b>       | End of loop                             |
| <b>C</b>       | Continue                                |

**Example (nested loop)**

| Command      | Description                 |
|--------------|-----------------------------|
| <b>L10</b>   | Loop 10 times               |
| <b>D4000</b> | Set distance to 4,000 steps |
| <b>G</b>     | Go                          |
| <b>L5</b>    | Loop 5 times                |
| <b>D10</b>   | Set distance to 10 steps    |
| <b>G</b>     | Go                          |
| <b>N</b>     | End of the 5 x 10 step      |
| <b>N</b>     | End of the overall loop     |

The commands L5 D10 G N form a nested loop.

|           |                           |                                  |
|-----------|---------------------------|----------------------------------|
| <b>LA</b> | <b>Limit Deceleration</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|---------------------------|----------------------------------|

| SYNTAX | UNITS                 | RANGE                  | DEFAULT | ATTRIBUTES                                  |
|--------|-----------------------|------------------------|---------|---|
| <a>LAn | Revs/sec <sup>2</sup> | n = 0.001 -<br>999,999 | 900     | Buffered, Universal,<br>Savable in Sequence |

**Description** Define or report the deceleration rate after limit switch operation. This allows a rapid deceleration in response to a limit switch regardless of the rate programmed by the "A" command.

|           |                      |                                  |
|-----------|----------------------|----------------------------------|
| <b>LD</b> | <b>Limit Disable</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|----------------------|----------------------------------|

| SYNTAX | UNITS | RANGE     | DEFAULT | ATTRIBUTES                                  |
|--------|-------|-----------|---------|---|
| <a>LDn | N/A   | n = 0 - 3 | 0       | Buffered, Universal,<br>Savable in Sequence |

**Description** This command is used to disable the limit switch functions.

n = 0: Enable all limits (default)  
n = 1: Disable limit +  
n = 2: Disable - limit  
n = 3: Disable + & - limit

**Example**

| <u>Command</u> | <u>Description</u>  |
|----------------|---|
| <b>1LD3</b>    | The detection of both + and - limit switch operation is disabled for axis 1 |

|           |  |                                  |
|-----------|--|----------------------------------|
| <b>LS</b> | <b>Limit Switch Fast Deceleration Stop</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|--|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                           |
|--------|-------|-------|---------|--------------------------------------|
| <a>LS  | N/A   | N/A   | N/A     | Immediate, Universal,<br>Never Saved |

**Description** Decelerate and stop at the limit switch deceleration rate set by LA (usually set fast compared to normal acceleration rate).

**MAS****Mode Asynchronous**

Not used in PS7X, BLX

**MC****Mode Continuous****VALID FROM**  
Version 1.4**SYNTAX**

&lt;a&gt;MC

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**Buffered, Universal,  
Savable in Sequence**EXECUTION TIME****SEE ALSO** MN, A, V**Description**

MC sets the move mode to continuous. It causes subsequent moves to ignore any distance parameter and move continuously at the programmed velocity until stopped by an S, LS or K command.

**Example**Command**MC****H-****A10****V10****G**Description

Set continuous mode

Set direction to negative

Set acceleration to 10 rev/sec/sec

Set velocity to 10 rev/sec

Go - run continuously at 10 rev/sec

**MN****Mode Normal****VALID FROM**  
Version 1.4**SYNTAX****UNITS****RANGE****DEFAULT****ATTRIBUTES**

<a>MN      N/A                      N/A                      N/A                      Buffered, Universal, Savable in Sequence

**EXECUTION TIME**

**SEE ALSO**    MQ, MC, MPA, MPI, A, V, D

**Description**      MN sets the move mode to normal preset distance. It causes last issued distance parameter to be used as the distance for the current move. The MN command will change the mode of operation from continuous back to preset.

**Example**

| <u>Command</u> | <u>Description</u>                 |
|----------------|------------------------------------|
| <b>MN</b>      | Set mode to normal                 |
| <b>A10</b>     | Set acceleration to 10 rev/sec/sec |
| <b>V10</b>     | Set velocity to 10 rev/sec         |
| <b>D8000</b>   | Set distance to 8000 steps         |
| <b>G</b>       | Go - run for 8000 steps            |

|            |                               |                                  |
|------------|-------------------------------|----------------------------------|
| <b>MPA</b> | <b>Mode Position Absolute</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-------------------------------|----------------------------------|

|               |              |              |                |  |
|---------------|--------------|--------------|----------------|--|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                        |
| <a>MPA        | N/A          | N/A          | N/A            | Buffered, Universal, Savable in Sequence |

**EXECUTION TIME**

**SEE ALSO**    MN, MQ, MC, MPI, D, PZ

**Description**      Sets the position mode to absolute. In this mode all move distances are referenced to absolute zero. Units are scaled by the CUR command. You must be in the preset mode (MN or MQ command) before the MPA command will take effect.

**Example**

| <u>Command</u> | <u>Description</u>                               |
|----------------|--|
| <b>MN</b>      | Set normal mode                                  |
| <b>MPA</b>     | Set absolute mode                                |
| <b>A10</b>     | Set acceleration to 10 revs/sec <sup>2</sup>     |
| <b>V10</b>     | Set velocity to 10 revs/sec                      |
| <b>D10000</b>  | Set new absolute position to 10,000 steps        |
| <b>G</b>       | Go - move to position 10,000                     |
| <b>D2000</b>   | Set new absolute position to 2,000 steps         |
| <b>G</b>       | Go - move 8,000 steps negative to position 2,000 |

Note that in the absolute mode, giving two G (go) commands in succession will cause the motor to move only once as the motor will have achieved the desired absolute position at the end of the first move.

|            |                                  |                                  |
|------------|----------------------------------|----------------------------------|
| <b>MPI</b> | <b>Mode Position Incremental</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|----------------------------------|----------------------------------|

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                        |
|---------------|--------------|--------------|----------------|--|
| <a>MPI        | N/A          | N/A          | N/A            | Buffered, Universal, Savable in Sequence |

**EXECUTION TIME** **SEE ALSO** MPA, D

**Description** Sets the position mode to incremental. In this mode all move distances are referenced to the starting position of each move. You must be in the Preset mode (MN or MQ command) before the MPI command will take effect. MPI is the power up default on the MN command.

**Example**

| <u>Command</u> | <u>Description</u>   |
|----------------|--|
| <b>MN</b>      | Set normal mode  |
| <b>MPI</b>     | Set incremental mode   |
| <b>A10</b>     | Set acceleration to 10 revs/sec <sup>2</sup>                 |
| <b>V10</b>     | Set velocity to 10 revs/sec                                  |
| <b>D8000</b>   | Set distance to 8000 steps                                   |
| <b>G</b>       | Go - move 8000 steps positive                                |
| <b>D-4000</b>  | Set distance to 4000 steps negative                          |
| <b>G</b>       | Go - move to position +4000 steps from the starting position |

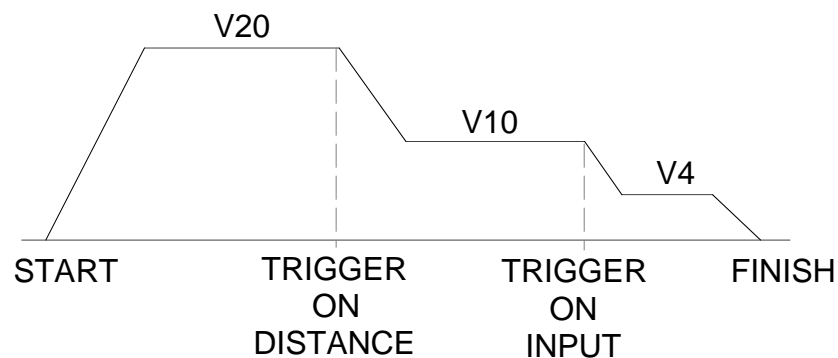
|           |                          |  |  |                                  |
|-----------|--------------------------|--|--|----------------------------------|
| <b>MQ</b> | <b>Speed Change Mode</b> |  |  | <b>VALID FROM</b><br>Version 1.4 |
|-----------|--------------------------|--|--|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                                  |
|--------|-------|-------|---------|---|
| <a>MQ  | N/A   | N/A   | N/A     | Buffered, Universal,<br>Savable in Sequence |

**Description** This command allows velocity to be changed using buffered commands during a preset move.

**Example**

| <u>Command</u> | <u>Description</u>                           |
|----------------|--|
| <b>MQ</b>      | Set speed change mode                        |
| <b>A50</b>     | Set acceleration to 50 revs/sec <sup>2</sup> |
| <b>V20</b>     | Set speed to 20 revs/sec                     |
| <b>D8000</b>   | Set total distance to 8000 steps             |
| <b>G</b>       | Go - start move                              |
| <b>TRD3000</b> | Change speed at distance of 3000 steps       |
| <b>V10</b>     | New speed set to 10 revs/sec                 |
| <b>TRE_X1</b>  | Change speed when input goes to 1            |
| <b>V4</b>      | New speed set to 4 revs/sec                  |



**Figure 5-8. Complex Velocity Profile Using MQ Mode**

The motor accelerates at 50 revs/sec<sup>2</sup> to a velocity of 20 revs/sec. At a distance of 3000 steps the velocity changes to 10 revs/sec. When Input 2 is energised the velocity changes to 4 revs/sec until the programmed distance is reached at 8000 steps.

**MSY****Mode Synchronous****Not used in PS7X, BLX****N****End Loop****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;N

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**Buffered, Universal,  
Savable in Sequence**EXECUTION TIME****SEE ALSO** L, Y**Description**

Marks the end of a loop. When used in conjunction with the L command, it causes the buffered commands between the L and the N to be executed as many times as the number following L.

**Example**Command**L5****A10****V10****D8000****G****N**Description

Set to loop 5 times

Set acceleration to 10 revs/sec<sup>2</sup>

Set speed to 10 revs/sec

Set distance to 8000 steps

Go - move 8000 steps (repeated 5 times)

End of loop

**O****Programmable Output****VALID FROM**

Version 2.1

| SYNTAX       | UNITS | RANGE       | DEFAULT | ATTRIBUTES                               |
|--------------|-------|-------------|---------|--|
| <a>Obbb<bbb> | N/A   | b = 0, 1, X | N/A     | Buffered, Universal, Savable in Sequence |

**EXECUTION TIME****SEE ALSO** SS, OS**Description**

Turns outputs on and off. When HSD0 is set, no daughter board is fitted and the three outputs are set by the first three digits 'bbb' where the first 'b' sets output 1, the second 'b' output 2 and the third 'b' output 3. When HSD1 is set, the three additional outputs on the daughter board are set by the last three digits <bbb>. The fourth 'b' sets output 4, the fifth 'b' sets output 5 and the sixth 'b' sets output 6.

b = 1 sets the output to sink current, i.e. output on  
 b = 0 turns the output current off  
 b = X leaves output unchanged

**Examples**Command**1HSD1****1O111111****O000XXX****OX1XX0X**Description

Optional motherboard fitted

Outputs 1-6 on device 1 to sink current from an externally supplied source

Current stops flowing in outputs 1-3

Leaves outputs one, three, four and six unchanged while turning output 2 on and output 5 off

Using the configuration switches it is possible to dedicate output 1 as the "Composite fault" line and/or output 2 as an "in position" line. The O command will not control these outputs if they have been configured in this way.

**OFF****De-Energise Drive****VALID FROM**

Version 1.4

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES           |
|--------|-------|-------|---------|----------------------|
| <a>OFF | N/A   | N/A   | N/A     | Immediate, Universal |

**EXECUTION TIME****SEE ALSO** ST, ON**Description**

De-energises the drive immediately. This command may be used to shut down the drive quickly in an emergency. Issuing an SV command after the OFF command will cause the drive to power up in the de-energised state.

**ON****Energise Drive****VALID FROM**



|            |                                  |                                  |
|------------|----------------------------------|----------------------------------|
| <b>OSB</b> | <b>Integral Action Selection</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|----------------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                |
|--------|-------|------------|---------|---------------------------|
| aOSBn  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific |

**Description** This command selects whether integral action will occur all the time or only whilst in position

n = 1: Integral action will only occur whilst 'in-position'  
n = 0: Integral action occurs all the time

|            |                                  |                                  |
|------------|----------------------------------|----------------------------------|
| <b>OSC</b> | <b>Monitor Command Reporting</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|----------------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                |
|--------|-------|------------|---------|---------------------------|
| aOSCn  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific |

**Description** The OSC command selects continuous reporting of monitor commands (DS, DPA etc.) or reporting of one value only.

n = 1: Signal monitor commands report only one value  
n = 0: Monitor commands report continuously

|            |                                |                                  |
|------------|--------------------------------|----------------------------------|
| <b>OSD</b> | <b>Input 7 Sequence Select</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|--------------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                |
|--------|-------|------------|---------|---------------------------|
| aOSDn  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific |

**Description** This command configures Input 7 as a sequence select line enabling up to 16 sequence selections.

n = 1: Input 7 defined as sequence select  
n = 0: Input 7 not sequence select

|            |                   |                                  |
|------------|-------------------|----------------------------------|
| <b>OSE</b> | <b>Jog Enable</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-------------------|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                |
|--------|-------|------------|---------|---------------------------|
| aOSEN  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific |

**Description** The OSE command is used to enable or disable the jog function.

n = 1: Jog enabled  
n = 0: Jog disabled

**OSF****Initialisation On  
Limit****NOT USED IN BLX****OSG****Dither Enable****Not used in System 7,  
PS7X, BLX****OSH****Select Triangular/  
Square Dither****Not used in System 7,  
PS7X, BLX**

**OSI****Initialisation  
Movement****NOT USED IN BLX****OSJ****Select 24/16 Bit  
Gearbox****VALID FROM**  
Version 2.3**SYNTAX****UNITS****RANGE****DEFAULT****ATTRIBUTES**

aOSJn

N/A

n = 0 or 1

0

Buffered, Device Specific

**EXECUTION TIME:****SEE ALSO: RAT****Description**

The OSJ command selects 24 or 16 bit gearbox used with the RAT command for external pulse stream following.

- n = 1: 24 bit gearbox operation -  
RAT command range = +/-16777216
- n = 0: 16 bit gearbox operation  
RAT command range = +/-65535

The 24 bit range has been included in this software version for higher resolution, but the 16 bit range has been retained for compatibility with earlier software versions.

**OSK****Integrity Check of  
Optical Encoder****VALID FROM**  
Version 3.1**SYNTAX****UNITS****RANGE****DEFAULT****ATTRIBUTES**

aOSKn

N/A

n = 0 or 1

0

Buffered, Device Specific

**SEE ALSO CIX****Description**

This command enables an encoder integrity check for servo systems fitted with an incremental encoder.

- n = 1 The encoder count will be checked at the index track
- n = 0 The encoder count will not be checked

If a count error occurs at the index track with the OSK option enabled, the interface will de-energise the drive, and an RSE command will produce the message:

#90 Shutdown by <X> counts encoder error at index track

Bit 9 in the de-energise data will be set and the LED flash code will be set to 1.

The RE command will show the drive to be de-energised. This option requires the encoder to have an index pulse that is one pulse wide (when gated with the A and B encoder channels). The option also requires only one index pulse every time the actuator moves the number of pulses defined in the CMR command, i.e. generally, only one pulse per rev.

| <b>OSM</b>         |  | <b>Integral Action Sensitivity</b> |                | <b>VALID FROM</b><br>Version 3.1                 |
|--------------------|--|------------------------------------|----------------|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                       | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                                |
| aOSMn              | N/A  | n = 0 or 1                         | 0              | Buffered, Device Specific<br><b>SEE ALSO</b> CIG |
| <b>Description</b> | <p>This command enables a fast, wide range version of the integral action capability.</p> <p>n = 0 Accumulate at 20ms sampling to standard range<br/>n = 1 Accumulate at 2ms sampling to a 256 times larger range</p> <p>The default setting OSM0 is suitable for the majority of situations in which integral action would be needed, i.e. systems with a significant frictional load. Lightly-loaded systems using smaller motors may benefit from the OSM1 setting.</p> |                                    |                |  |

**OSO****Suppress Units****VALID FROM**  
Version 3.4**SYNTAX**

aOSOn

**UNITS**

N/A

**RANGE**

n = 0 or 1

**DEFAULT**

0

**ATTRIBUTES**

Buffered, Device Specific

**SEE ALSO** CMR, CUR,  
CIX**Description**

This command suppresses any reference to units such as RPS in the message prompts. This avoids confusion on linear and rotary systems where the velocity is not expressed in revolutions at a motor shaft.

n = 0 rps and rps/s units appear in messages

n = 1 rps and rps/s units do not appear in messages

**P****Position****VALID FROM**  
Version 1.4**SYNTAX**

aP

**UNITS**

User steps

**RANGE**

±2,097,151

**DEFAULT**

N/A

**ATTRIBUTES**

Immediate, Device Specific

**Description**

Displays the position relative to the start of the last index. The position counter is cleared and restarted by next G command. This is a single response in user steps, and can be used during a move.

**PIC****Picture****VALID FROM**  
Version 1.4**SYNTAX**

aPIC

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**

Immediate, Universal

**Description**

Displays a picture of the servo loop with the signal monitor numbers. A device address is not necessary since any control module will return the same display.

**PR****Position Report****VALID FROM**  
Version 1.5**SYNTAX**

aPR

**UNITS**

User steps

**RANGE**-268,435,455  
to +268,435,455**DEFAULT**

N/A

**ATTRIBUTES**

Buffered, Device Specific

**EXECUTION TIME****SEE ALSO** MPA, MPI,

D, PZ, SP

**Description** Requests the current absolute position. The report is a number preceded by a sign and followed by a carriage return \*(s)n[CR]. The number represents the cumulative position in user steps (n) with respect to the position at power up or the last point at which an SP or PZ command was issued resetting absolute zero. The sign indicates which side of the absolute zero position the motor is on.

**Example**

|                |  |
|----------------|--|
| <u>Command</u> | <u>Response</u>                                    |
| 1PR            | *-25600[CR] (Motor is at absolute position -25600) |

|           |              |                                  |
|-----------|--------------|----------------------------------|
| <b>PS</b> | <b>Pause</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|--------------|----------------------------------|

|               |              |              |                |  |
|---------------|--------------|--------------|----------------|--|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                        |
| <a>PS         | N/A          | N/A          | N/A            | Buffered, Universal, Savable in Sequence |

**EXECUTION TIME** **SEE ALSO** C, SSB

**Description** This command pauses the execution of the current command string or sequence. Execution will then be resumed after a C command (continue) is received. The command is used to allow the entering of a complete command string before the commands are executed.

**Example**

|                |  |
|----------------|--|
| <u>Command</u> | <u>Description</u>                         |
| <b>PS</b>      | Pause until C command is received          |
| <b>D5000</b>   | Set distance to 5000 steps                 |
| <b>G</b>       | Go - start move                            |
| <b>H-</b>      | Set direction to negative                  |
| <b>G</b>       | Go - start negative direction move         |
| <b>C</b>       | Release pause - both moves now carried out |

**PSY****Pause for  
Synchronisation**

Not used in PS7X, BLX

**PZ****Position Zero****VALID FROM**  
Version 1.4**SYNTAX****UNITS****RANGE****DEFAULT****ATTRIBUTES**

&lt;a&gt;PZ

N/A

N/A

N/A

Buffered, Universal,  
Savable in Sequence**EXECUTION TIME****SEE ALSO** MN, MPA,  
MPI, PR, D, SP**Description**

Sets the current position to be absolute zero.

**Example**CommandDescription**MPA**

Set absolute mode

**D2500**

Set new position to 2500 steps

**G**

Go - move to absolute position +2500

**1DPA**

Report the new position (+2500 returned)

**1PZ**

Set the new position as absolute zero

**1DPA**

Report the position (zero is returned)

Where a servo is used, due to offsets in the loop the demanded position and the feedback position may not be exactly equal. This command sets the physical or feedback position zero, so the demand position indicated by DPS may show a non-zero value

|           |                               |                                  |
|-----------|-------------------------------|----------------------------------|
| <b>QS</b> | <b>Transmit An Identifier</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-------------------------------|----------------------------------|

|               |              |              |                |   |
|---------------|--------------|--------------|----------------|---|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>QS         | N/A          | N/A          | N/A            | Buffered, Universal,<br>Savable in Sequence |

**Description** Responds with the identity of the echoing device. The command can be used to confirm which device is echoing after a series of universal commands.

The response consists of an asterisk, a two digit device address, an exclamation mark and a carriage return.

**Example**

| <u>Command</u> | <u>Description</u>                            |
|----------------|---|
| <b>MN</b>      | Select normal mode                            |
| <b>D4000</b>   | Set distance to 4000 steps                    |
| <b>A100</b>    | Set acceleration to 100 revs/sec <sup>2</sup> |
| <b>V50</b>     | Set velocity to 50 revs/sec                   |
| <b>G</b>       | Go  |
| <b>QS</b>      | Transmit identifier                           |

If this is executed as a sequence by axis 1, the response \*01! will be received over the RS232C line when the sequence is complete.

|          |                                       |                                  |
|----------|---------------------------------------|----------------------------------|
| <b>R</b> | <b>Report Serial Interface Status</b> | <b>VALID FROM</b><br>Version 2.6 |
|----------|---------------------------------------|----------------------------------|

XSR,XSS,DFS,  
S,DFX  
**Description**

Requests the status of the RS232C Control Module. The response is \*<char>[CR] :

|         |  |
|---------|--|
| *R [CR] | - ready for a command with no errors.                                  |
| *S [CR] | - ready for a command with function errors.                            |
| *T [CR] | - ready for a command with previous comms error.                       |
| *U [CR] | - ready for a command with function error and previous comms error.    |
| *B [CR] | - busy performing a move with no errors.                               |
| *C [CR] | - busy performing a move with function error.                          |
| *D [CR] | - busy performing a move with previous comms error.                    |
| *E [CR] | - busy performing a move with function error and previous comms error. |

In terminal mode a message appears as well. The ready response means a buffered command will be executed immediately on receipt.

The RS232 communications is considered to have an error if:

- A framing error occurred (start / stop bits incorrect)
- An overrun error occurred (new character received whilst previous one unread)

Reading the status clears an outstanding communications error.

**Example**

|                      |   |
|----------------------|---|
| <u>Command</u><br>1R | <u>Response</u><br>*R[CR] RS232C Control Module ready to accept a command. If it's a buffered command, it will be executed immediately. |
|----------------------|---|

|           |  |                                  |
|-----------|--|----------------------------------|
| <b>RA</b> | <b>Report A - Limit Status Request</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|--|----------------------------------|

|               |              |              |                |                            |
|---------------|--------------|--------------|----------------|----------------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>          |
| aRA           | N/A          | N/A          | N/A            | Immediate, Device Specific |

**Description** Requests the status of the limits. The response, in accordance with the following table, is the current and last move limit status represented by a single character from @[CR] to O[CR].

**Example**

|                       |  |
|-----------------------|--|
| <u>Command</u><br>1RA | <u>Response</u><br>*I [CR] The current move is limited by the negative limit and the last move was terminated by the positive limit. |
|-----------------------|--|

| Response Character | Last Move Terminated By |                | Current Move Limited By |                |
|--------------------|-------------------------|----------------|-------------------------|----------------|
|                    | Positive Limit          | Negative Limit | Positive Limit          | Negative Limit |
| *@                 | NO                      | NO             | NO                      | NO             |
| *A                 | YES                     | NO             | NO                      | NO             |
| *B                 | NO                      | YES            | NO                      | NO             |
| *C                 | YES                     | YES            | NO                      | NO             |
| *D                 | NO                      | NO             | YES                     | NO             |
| *E                 | YES                     | NO             | YES                     | NO             |
| *F                 | NO                      | YES            | YES                     | NO             |
| *G                 | YES                     | YES            | YES                     | NO             |
| *H                 | NO                      | NO             | NO                      | YES            |
| *I                 | YES                     | NO             | NO                      | YES            |
| *J                 | NO                      | YES            | NO                      | YES            |
| *K                 | YES                     | YES            | NO                      | YES            |
| *L                 | NO                      | NO             | YES                     | YES            |
| *M                 | YES                     | NO             | YES                     | YES            |
| *N                 | NO                      | YES            | YES                     | YES            |
| *O                 | YES                     | YES            | YES                     | YES            |

The RA command is useful when the motor will not move in either or both directions. The report back will indicate whether or not the last move was terminated by a limit switch activation and if the current move is disabled by an active limit.

|            |                                  |                                  |
|------------|----------------------------------|----------------------------------|
| <b>RAT</b> | <b>Set Rate Multiplier Value</b> | <b>VALID FROM</b><br>Version 2.4 |
|------------|----------------------------------|----------------------------------|

| <b>SYNTAX</b> | <b>UNITS</b>          | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>             |
|---------------|-----------------------|-----------------------|----------------|-------------------------------|
| aRATn         | Rate multiplier value | See description       | 16384          | Buffered, Device Specific     |
|               |                       | <b>EXECUTION TIME</b> |                | <b>SEE ALSO</b> CCS, SIM, OSJ |

**Description**

This command is used in servo mode (HSC1) to configure a rate multiplier on the second encoder input when using the following mode of operation. The range of the command is determined by the setting of the OSJ command as follows:

OSJ = 0: RAT range -65536 to +65536

OSJ = 1: RAT range -16777216 to +16777216

The effect is that of a gearbox on the second encoder input that obeys the following formula:

Output rate to the motor (OSJ = 0) =  $\frac{n}{65536}$  x input rate from the second encoder.

Output rate to the motor (OSJ = 1) =  $\frac{n}{16777216}$  x input rate from the second encoder.

For a limited number of pulses:

Pulses to motor (OSJ = 0) = INT  $\frac{n \times \text{pulses in}}{65536}$

Pulses to motor (OSJ = 1) = INT  $\frac{n \times \text{pulses in}}{16777216}$

where INT means 'the integer part of'. The value is rounded down towards zero whether the value is positive or negative.

Negative values of RAT allow the direction of following to be reversed.

**NOTE:** The number of second encoder input pulses is also affected by the CCS command.

**Example** A 4000 step/rev motor is required to follow a 1000 line/rev encoder at half the encoder speed in the opposite direction. 1 revolution of the encoder produces 1000 x 4 (CCS0 set) = 4000 pulses; -2000 pulses are required to produce a half revolution of the motor shaft in the opposite direction.

$$\text{So } n = \frac{65536 \times (-2000)}{4000} = -32768$$

| <u>Command</u>    | <u>Description</u>   |
|-------------------|--|
| <b>1CCS0</b>      | Select x 4 decode  |
| <b>1OSJ0</b>      | Set OSJ for 65536 RAT range  |
| <b>1SIM1</b>      | Select encoder following mode  |
| <b>1RAT-32768</b> | If the tracked encoder turns 1 revolution CW, the number of pulses sent to the motor is: |

$$\frac{-32768 \times 4000}{65536} = -2000$$

The motor of axis 1 therefore turns a half revolution CCW.

|           |  |                                  |
|-----------|--|----------------------------------|
| <b>RB</b> | <b>Report B - Misc. Status Request</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|--|----------------------------------|

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>             |
|---------------|--------------|-----------------------|----------------|-------------------------------|
| aRB           | N/A          | N/A                   | N/A            | Immediate, Device Specific    |
|               |              | <b>EXECUTION TIME</b> |                | <b>SEE ALSO</b> TR, PS, L, ST |

**Description** Requests the status of a loop, a pause, a shutdown or an input . The response is @ [CR] to O [CR] according to the following table.

| Response Character | Loop Active | Pause Active | Shutdown Active | Input Active |
|--------------------|-------------|--------------|-----------------|--------------|
| *@                 | NO          | NO           | NO              | NO           |
| *A                 | YES         | NO           | NO              | NO           |
| *B                 | NO          | YES          | NO              | NO           |
| *C                 | YES         | YES          | NO              | NO           |
| *D                 | NO          | NO           | YES             | NO           |
| *E                 | YES         | NO           | YES             | NO           |
| *F                 | NO          | YES          | YES             | NO           |
| *G                 | YES         | YES          | YES             | NO           |
| *H                 | NO          | NO           | NO              | YES          |
| *I                 | YES         | NO           | NO              | YES          |
| *J                 | NO          | YES          | NO              | YES          |
| *K                 | YES         | YES          | NO              | YES          |
| *L                 | NO          | NO           | YES             | YES          |
| *M                 | YES         | NO           | YES             | YES          |
| *N                 | NO          | YES          | YES             | YES          |
| *O                 | YES         | YES          | YES             | YES          |

Loop active means that a loop is in progress.

Pause active means that buffered commands are not being executed and the RS232C Control Module is waiting for a C command.

Shutdown active means that the motor is shutdown.

Input active means that at least one input is active.

#### Example

Command  
1RB

Response  
\*J[CR] There is no loop active, pause is active, the motor is not shutdown and there is an input active.

|           |                             |                                  |
|-----------|-----------------------------|----------------------------------|
| <b>RE</b> | <b>Drive Status Request</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-----------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                 |
|--------|-------|-------|---------|----------------------------|
| aRE    | N/A   | N/A   | N/A     | Immediate, Device Specific |

**Description** Requests the energised/de-energised status of the drive. The response is \*@ if the drive is energised or \*B if the drive is de-energised.



|           |                              |                                  |
|-----------|------------------------------|----------------------------------|
| <b>RG</b> | <b>Report Go Home Status</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|------------------------------|----------------------------------|

|               |              |              |                |                            |
|---------------|--------------|--------------|----------------|----------------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>          |
| aRG           | N/A          | N/A          | N/A            | Immediate, Device Specific |

**EXECUTION TIME****SEE ALSO** GH, GA, GHP, GHF**Description**

Requests the status of the last Go Home attempt. The response is @ [CR] or A [CR] , indicating the success or failure of last go home attempt as follows:

| Response | Go Home Successful |
|----------|--------------------|
| *@       | NO                 |
| *A       | YES                |

|             |   |                                  |
|-------------|---|----------------------------------|
| <b>RIFS</b> | <b>Return Indexer to Factory Settings</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|---|----------------------------------|

|               |              |              |                |                     |
|---------------|--------------|--------------|----------------|---------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>   |
| <a>RIFS       | N/A          | N/A          | N/A            | Immediate Universal |

**EXECUTION TIME****SEE ALSO** Z, RFS**Description**

This command sets indexer default values appropriate to the product number entered using the RFS command.

**Example**

Command  
1RFS2  
1RIFS

Description  
Set axis 1 as a System 7 microstepper  
Set indexer default values suitable for System 7 microstepper

|            |                             |                                  |
|------------|-----------------------------|----------------------------------|
| <b>RPO</b> | <b>Report Power-On Time</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-----------------------------|----------------------------------|

|               |              |              |                |                            |
|---------------|--------------|--------------|----------------|----------------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>          |
| aRPO          | N/A          | N/A          | N/A            | Immediate, Device Specific |

**Description** Requests the RS232C Control Module power on time (hours) in decimal, for example 33.7 hours. A continuous record of usage is maintained by the interface.

|           |                               |                                  |
|-----------|-------------------------------|----------------------------------|
| <b>RS</b> | <b>Report Sequence Status</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-------------------------------|----------------------------------|

|               |              |              |                |                            |
|---------------|--------------|--------------|----------------|----------------------------|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>          |
| aRS           | N/A          | N/A          | N/A            | Immediate, Device Specific |

**EXECUTION TIME** **SEE ALSO** XR, XP

**Description** Request the status of the last sequence execution. The response will be @ [CR] to D [CR]) according to the following table:-

| Response Character | Sequence Started | Sequence Ended |
|--------------------|------------------|----------------|
| *@                 | NO               | NO             |
| *A                 | YES              | NO             |
| *B                 | NO               | YES            |
| *C                 | YES              | YES            |
| *D                 | NO               | YES            |

Whenever a sequence is started, the sequence start bit is set and the sequence end bit is cleared (this only occurs if the sequence is valid and is actually run). Whenever a sequence is ended, the start bit is cleared and the end bit is set. Any abrupt move termination (e.g. limit activation), or a K or S command clears both bits.

\*D is reported when there is an unbalanced number of loops and loop terminators inside a sequence. Starting a loop in one sequence and terminating it in another sequence is not allowed. Nested loops require complete closure before execution will begin.

**Example**

|                |                         |
|----------------|-------------------------|
| <u>Command</u> | <u>Response</u>         |
| 1RS            | *A Sequence in progress |

**RSE****Report Servo Errors**
**VALID FROM**  
Version 3.0

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>          |
|---------------|--------------|--------------|----------------|----------------------------|
| aRSE          | N/A          | N/A          | N/A            | Immediate, Device Specific |

**Description**

Returns all servo error flags as a set of messages to the terminal and as a bit pattern. The messages correspond with the drive Fault LED indications. The bit pattern sent is:-

\*xxxx\_xxxx\_000x\_xxxx\_xxxx where the rightmost "x" bit is bit 0.

Possible messages include:

| <b>Message</b> | <b>Bit No.</b> | <b>Flash Code</b> | <b>Fault</b>  |
|----------------|----------------|-------------------|---|
| #20            | 0              | 1                 | De-energised by ST1 or OFF command                  |
| #22            | 2              | 3                 | EPROM changed with different memory map             |
| #23            | 3              | 4                 | Excessive position error                            |
| #24            | 4              | 5                 | Memory failure - failed checksum                    |
| #25            | 5              | 1                 | Undefined drive                                     |
| #26            | 6              | 7                 | Prolonged max torque demand                         |
| #27            | 7              | 8                 | Output to drive is zero torque                      |
| #28            | 8              | 8                 | Emergency stop input seen                           |
| #29            | 1              | 1                 | Shutdown by limit switch during initialisation      |
| #90            | 9              | 1                 | Shutdown by <n> counts encoder error at index track |
| #80            | 12             | 2                 | Drive disabled by composite drive fault             |
| #81            | 13             | 2                 | Impending power loss                                |
| #82            | 14             | 2                 | Drive disabled by motor over temperature            |
| #83            | 15             | 2                 | Drive disabled by drive over temperature            |
| #84            | 16             | 2                 | Drive disabled by HT over voltage                   |
| #85            | 17             | 2                 | Drive disabled by drive over current                |
| #86            | 18             | 2                 | Drive disabled by drive specific fault (see manual) |
| #87            | 19             | 2                 | Drive disabled by drive logic supply fault          |

|           |                       |                                  |
|-----------|-----------------------|----------------------------------|
| <b>RV</b> | <b>Revision Level</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|-----------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                 |
|--------|-------|-------|---------|----------------------------|
| aRV    | N/A   | N/A   | N/A     | Immediate, Device Specific |

**Description** Request software version number. An example response is:

\* ISSUE: 1.1 DATE: 27-Jun-89 17:29 [CR]

This is the software revision and the date and time of the revision.

|          |             |                                  |
|----------|-------------|----------------------------------|
| <b>S</b> | <b>Stop</b> | <b>VALID FROM</b><br>Version 1.4 |
|----------|-------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES           |
|--------|-------|-------|---------|----------------------|
| <a>S   | N/A   | N/A   | N/A     | Immediate, Universal |

#### EXECUTION TIME

**SEE ALSO** K

#### Description

Stop. If SSH 0 (Don't save command buffer on stop) is set, the command buffer is cleared (at the end of a move if one is in progress).

A command sequence download is aborted (XD command) and a time delay is terminated.

The motor is decelerated to stop using the current acceleration value ('A') but the drive stays energised.

If SSH1 (Save command buffer on stop) is set, the command buffer is not cleared and only the move that is in progress is terminated.

#### Example

##### Command

**MC**

**A10**

**V10**

**G**

**S**

##### Description

Set continuous mode

Set acceleration to 10 revs/sec<sup>2</sup>

Set velocity to 10 revs/sec

Go - run continuously

The motor will decelerate to a stop at a rate of 10 revs/sec/sec as soon as S is entered

|             |             |                                  |
|-------------|-------------|----------------------------------|
| <b>SAVE</b> | <b>Save</b> | <b>VALID FROM</b><br>Version 1.4 |
|-------------|-------------|----------------------------------|

| SYNTAX  | UNITS | RANGE | DEFAULT | ATTRIBUTES                 |
|---------|-------|-------|---------|----------------------------|
| <a>SAVE | N/A   | N/A   | N/A     | Immediate, Device Specific |

**Description** This is an alternative expression for the SV command. It causes the current servo, set up and index parameters to be saved in non-volatile RAM.

Note that in order to successfully save data, link 1 must be in place. Remove the link to write protect the RAM.

|           |                      |                                  |
|-----------|----------------------|----------------------------------|
| <b>SB</b> | <b>Stop Buffered</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|----------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                               |
|--------|-------|-------|---------|--|
| <a>SB  | N/A   | N/A   | N/A     | Buffered, Universal, Savable in Sequence |

**Description** Stop. The motor is decelerated to a stop using the current acceleration value (A). The command can be used in the MC or MQ modes.

**Example**

| Command     | Description   |
|-------------|---|
| <b>2A10</b> | Set acceleration for axis 2 to 10 revs/sec <sup>2</sup>                 |
| <b>2SB</b>  | Axis 2 is decelerated to a stop at the rate of 10 revs/sec <sup>2</sup> |

|            |                                   |                                  |
|------------|-----------------------------------|----------------------------------|
| <b>SIM</b> | <b>Set Indexer/Following Mode</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-----------------------------------|----------------------------------|

| SYNTAX | UNITS           | RANGE | DEFAULT | ATTRIBUTES                |
|--------|-----------------|-------|---------|---------------------------|
| aSIMn  | n = Mode number | 0 - 5 | 0       | Buffered, Device Specific |

**EXECUTION TIME**

**SEE ALSO** CCS, RAT, FOL

**Description** This command is used to configure the serial control module for following a second encoder. The mode of operation is selected by n as follows:-

- n = 0 Normal control module operation. In this mode any pulses arriving at the second encoder input are ignored.
- n = 1 Encoder following operation. In this mode the position of the motor follows the pulse stream on the second encoder input and the control module motion commands are inactive.
- n = 2 Super position operation. In this mode both the incoming pulses from the second encoder and control module motion commands are active. The position of the motor is the sum of both motions.
- n = 3 Positive software scaled encoder following operation.
- n = 4 Negative software scaled encoder following operation.
- n = 5 Preset following index mode

**Note:** SIM1 and SIM2 can only be used in stepper mode with a 1:1 following ratio. The RAT command is not valid for stepper mode.

SIM3 and SIM4 provide a software scaled following capability which allows the input pulse stream to be multiplied or divided with a sign reversal if required. The scaling ratio is set by the CMR and CUR commands which define the motor and user resolutions. More information on encoder following can be found in Section H.

SIM5 selects indexing at a speed determined by the external input. The percentage following factor is set by the FOL command. More information on preset following index mode can be found in Section H.

**NOTE:** The stop command does not affect the second encoder input. Energise/de-energise commands are effective.

#### Example

| <u>Command</u> | <u>Description</u>            |
|----------------|-------------------------------|
| <b>1SIM0</b>   | Set normal control mode       |
| <b>GH2</b>     | Go home                       |
| <b>1SIM1</b>   | Set control by second encoder |

|            |                         |                                  |
|------------|-------------------------|----------------------------------|
| <b>SKE</b> | <b>Skip On 'Equals'</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-------------------------|----------------------------------|

| SYNTAX  | UNITS | RANGE | DEFAULT | ATTRIBUTES                                  |
|---------|-------|-------|---------|---|
| <a>SKEn | N/A   | N/A   | N/A     | Buffered, Universal,<br>Savable in Sequence |

**Description** Skip the following command on an input pattern equal to n. This command is used in sequences to control execution flow. The format for n, the input pattern, is specified by:

bbbbbbb\_bbbbbbb

The pattern is from left to right :

HOME, LMT-, LMT+, +JOG, -JOG, EIN (DRIVE ENERGISED), INDEX TRACK\_INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care). You can omit trailing X specifiers within each group of 7.

For example the following two pattern are equivalent:

001XXXX\_XXX11XX and

001\_XXX11

The next two are also equivalent:

XXXXXXXX\_X1XXXXX and

\_X1

**Example**

Command  
**1SKEXX1\_**

Description  
The next command will be skipped if the positive limit input is active

|            |                            |                                  |
|------------|----------------------------|----------------------------------|
| <b>SKN</b> | <b>Skip On 'Not Equal'</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|----------------------------|----------------------------------|

|               |              |              |                |  |
|---------------|--------------|--------------|----------------|--|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                        |
| <a>SKNn       | N/A          | N/A          | N/A            | Buffered, Universal, Savable in Sequence |

**Description** Skip the next command on input pattern not equal to n. The command is used in sequences to control execution flow. The format for n, the input pattern, is specified by:

bbbbbbb\_bbbbbbb

This pattern represents from left to right :

HOME, LMT-, LMT+, +JOG, -JOG, EIN (DRIVE ENERGISED), INDEX TRACK\_INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6, INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care). You can omit trailing X specifiers within each group of 7 specifiers.

For example the following two specifiers are equivalent:

001XXXX\_XXX11XX and

001\_XXX11

The next two are also equivalent:

XXXXXXXX\_X1XXXXX and

\_X1

**Example**

|                                 |  |
|---------------------------------|--|
| <u>Command</u><br><b>1SKN1_</b> | <u>Description</u><br>The next command will be skipped if the HOME input is not active |
|---------------------------------|--|

|           |                     |                                  |
|-----------|---------------------|----------------------------------|
| <b>SP</b> | <b>Set Position</b> | <b>VALID FROM</b><br>Version 2.2 |
|-----------|---------------------|----------------------------------|

|               |              |                                 |                |   |
|---------------|--------------|---------------------------------|----------------|---|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b>                    | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                                 |
| aSP<s>n       | Motor steps  | -268,435,455<br>to +268,435,455 | 0              | Buffered, Device Specific,<br>Savable in Sequence |
|               |              | <b>EXECUTION TIME</b>           |                | <b>SEE ALSO</b> MPA, MPI,<br>PR, D, PZ            |

**Description** Adds the value of n to the absolute position counter. The counter will be set to n if a PZ command is given prior to the SP command. This command is useful for setting the absolute zero point to some location other than that of the physical hardware home. If you have a cut-off saw for example, you may not be able to mount the home switch at the cut point. However, by mounting the home a known distance away and resetting the reference point with the SP command, the system may be made to function as if the home switch were at the cut point.

Note that the units of the SP command are scaled by the CMR command. You can get a position report in the same units by using the DPA command.

**Example**

|                |   |
|----------------|---|
| <u>Command</u> | <u>Description</u>  |
| <b>GH2</b>     | Go home   |
| <b>PZ</b>      | Set position to zero  |
| <b>SP4000</b>  | The absolute position will be set to 4000 so that the absolute zero will be one motor revolution away from the switch location. |

|           |                     |                                  |
|-----------|---------------------|----------------------------------|
| <b>SS</b> | <b>Set Switches</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|---------------------|----------------------------------|

|               |              |              |                |   |
|---------------|--------------|--------------|----------------|---|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                                 |
| aSS           | N/A          | N/A          | N/A            | Buffered, Device Specific,<br>Savable in Sequence |

**Description** The command SS will report the state of the SS switches with a response \*bbbb\_bbbb\_bbbb\_bbbb where the order of the bits is:  
  
ABCD\_EFGH\_I(X)(X)(X)\_(X)(X)(X)(X)  
  
(X) = not used. Therefore bit 1 indicates the status of SSA bit 2 the status of SSB etc.

|            |                            |                                  |
|------------|----------------------------|----------------------------------|
| <b>SSA</b> | <b>RS232C Echo Control</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|----------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                       |
|--------|-------|------------|---------|----------------------------------|
| aSSAn  | N/A   | n = 0 or 1 | 0       | Buffered,<br>Savable in Sequence |

**Description** This command turns the RS232C echo on and off

SSA0 = Echo on  
SSA1 = Echo off

In the Echo On mode characters that are received by the RS232C Control Module are echoed automatically. In the Echo Off mode, characters are not echoed. This command is useful if your computer cannot handle echoes.

**Example**

Command  
**SSA1**

Description  
Turns the echo off (Characters sent to the RS232C Control Module are not echoed back to the host).

|            |   |                                  |
|------------|---|----------------------------------|
| <b>SSB</b> | <b>Set Input 6 as a Clear Pause Input</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|---|----------------------------------|

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES  |
|--------|-------|------------|---------|---|
| aSSBb  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific,<br>Savable in Sequence |

**EXECUTION TIME**

**SEE ALSO** U, PS

**Description** This command can be used to allow input 6 to clear a pause. The input will function in the same way as issuing a 'C' command.

b =0: Normal  
b =1: Input 6 clears a pause

Note that input 6 should be low (i.e. off) when the PS command is issued, otherwise the command will be ignored and there will be no pause.

**Example**

Command  
**1SSB1**  
**1PS**

Description  
Set Input 6 as clear pause for axis 1  
This causes axis 1 to pause. Input 6 can be used to clear the pause.

|                        |   |                            |                     |  |
|------------------------|---|----------------------------|---------------------|--|
| <b>SSC</b>             | <b>Set Output 2 as an In-Position Output</b>  |                            |                     | <b>VALID FROM</b><br>Version 1.4                                       |
| <b>SYNTAX</b><br>aSSCb | <b>UNITS</b><br>N/A   | <b>RANGE</b><br>n = 0 or 1 | <b>DEFAULT</b><br>0 | <b>ATTRIBUTES</b><br>Buffered, Device Specific,<br>Savable in Sequence |
| <b>EXECUTION TIME</b>  |   |                            | <b>SEE ALSO</b>     | CEW, CIT   |
| <b>Description</b>     | b = 0: Normal<br>b = 1: Output 2 is configured as an 'in position' output<br>With SSC set to 1, output 2 will turn on when the motor is within the error window for the specified time defined by CEW & CIT commands. |                            |                     |  |

|                        |   |                            |                     |  |
|------------------------|---|----------------------------|---------------------|--|
| <b>SSD</b>             | <b>Set Output 1 as Composite Fault Signal</b>   |                            |                     | <b>VALID FROM</b><br>Version 2.2                                       |
| <b>SYNTAX</b><br>aSSDb | <b>UNITS</b><br>N/A   | <b>RANGE</b><br>n = 0 or 1 | <b>DEFAULT</b><br>0 | <b>ATTRIBUTES</b><br>Buffered, Device Specific,<br>Savable in Sequence |
| <b>Description</b>     | b = 0: Normal<br>b = 1: Output 1 is configured as a composite fault indicator<br>When functioning as a composite fault indicator, the output is normally on and any fault that de-energises the drive generated by software, watchdog timeout or an incoming emergency stop signal will cause the output to turn off.<br>This does not include the effect of ST1 or OFF, which deliberately de-energises the drive. If ST1 or OFF have been sent, the composite fault output will not then indicate the occurrence of an ESTOP input or of any other fault conditions such as excessive position error. |                            |                     |  |

|                        |  |                            |                     |  |
|------------------------|--|----------------------------|---------------------|--|
| <b>SSE</b>             | <b>Set Input 5 as a Controlled Stop Input</b>                            |                            |                     | <b>VALID FROM</b><br>Version 1.4                                       |
| <b>SYNTAX</b><br>aSSEb | <b>UNITS</b><br>N/A  | <b>RANGE</b><br>n = 0 or 1 | <b>DEFAULT</b><br>0 | <b>ATTRIBUTES</b><br>Buffered, Device Specific,<br>Savable in Sequence |
| <b>EXECUTION TIME</b>  |  |                            | <b>SEE ALSO</b>     | LA   |
| <b>Description</b>     | b = 0: Normal<br>b = 1: Input 5 is configured as a controlled stop input |                            |                     |  |

If the command SSE1 is given, a controlled stop at the limit deceleration rate (LA) will occur on an Input 5 transition to high level.

**Example**

| <u>Command</u> | <u>Description</u>  |
|----------------|---|
| <b>2LA10</b>   | Set limit deceleration to 10 revs/sec <sup>2</sup>  |
| <b>2A100</b>   | Set acceleration to 100 revs/sec <sup>2</sup>   |
| <b>2V5</b>     | Set velocity to 5 revs/sec  |
| <b>2SSE1</b>   | Set input 5 as controlled stop  |
| <b>2G</b>      | With axis 2 running at the programmed 5 revs/sec, Input 5 going to high level causes a stop at the programmed limit deceleration, 10 rev/sec <sup>2</sup> . |

## SSF

### Set Input 4 as Sequence Strobe

#### VALID FROM

Version 1.4

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                                     |
|--------|-------|------------|---------|--|
| aSSFb  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific, Savable in Sequence |

**Description** b = 0: Normal  
 b = 1: Input 4 defined as a sequence strobe. When active, Input 4 causes execution of the sequence number found at the value on Inputs 1-3, or inputs 1-3 and input 7 if OSD1 is selected.

See 'Sequence Programming'.

## SSG

### Save Command Buffer On Limit

#### VALID FROM

Version 1.4

| SYNTAX | UNITS | RANGE      | DEFAULT | ATTRIBUTES                                     |
|--------|-------|------------|---------|--|
| aSSGb  | N/A   | n = 0 or 1 | 0       | Buffered, Device Specific, Savable in Sequence |

**Description** b = 0: Normal  
 b = 1: Command buffer saved on limit

Normally, when a limit is hit, the current command buffer or sequence is scrapped thus preventing further execution. Setting this bit to 1 prevents this activity.

|                    |   |              |                |   |
|--------------------|---|--------------|----------------|---|
| <b>SSH</b>         | <b>Save Command Buffer On Stop</b>  |              |                | <b>VALID FROM</b><br>Version 1.4                  |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                                 |
| aSSHb              | N/A   | n = 0 or 1   | 0              | Buffered, Device Specific,<br>Savable in Sequence |
| <b>Description</b> | <p>b = 0: Normal<br/>b = 1: Save command buffer on stop ('S' command).</p> <p>This command operates in a similar way to SSG, but relates to the stop command.</p> |              |                |   |

|                    |   |              |                |                                  |
|--------------------|---|--------------|----------------|----------------------------------|
| <b>SSI</b>         | <b>Set All Inputs As Sequence Select</b>  |              |                | <b>VALID FROM</b><br>Version 3.1 |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                |
| aSSIn              | N/A   | n = 0 or 1   | 0              | Buffered, Device Specific,       |
| <b>Description</b> | <p>This command configures inputs 5,6, and 7 as sequence select lines enabling up to 63 sequence selections.</p> <p>n = 0 Inputs 5,6,7 not configured as sequence select lines<br/>n = 1 Inputs 5,6,7 are configured as sequence select lines</p> <p>See Standalone operation at the beginning of this Section.</p> |              |                |                                  |

|                    |   |              |                |   |
|--------------------|---|--------------|----------------|---|
| <b>ST</b>          | <b>Shutdown</b>   |              |                | <b>VALID FROM</b><br>Version 1.4            |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>STn             | N/A   | n = 1 or 0   | N/A            | Buffered, Universal,<br>Savable in Sequence |
| <b>Description</b> | <p>Motor shutdown.</p> <p>0: The drive is enabled and any brushless or stepper motor initialised if required.<br/>1: The drive is disabled.</p> |              |                |   |





Note that in order to successfully save data, link 1 must be in place. Remove the link to write protect the RAM.

| <b>T</b>           |  | <b>Time Delay</b> |   | <b>VALID FROM</b><br>Version 1.4         |
|--------------------|--|-------------------|---|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>      | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                        |
| <a>Tn              | Seconds  | n = 0.01 - 999.99 | N/A   | Buffered, Universal, Savable in Sequence |
| <b>Description</b> | This command causes the RS232C Control Module to wait the number of seconds specified before it executes the next command in the buffer. |                   |   |  |
| <b>Example</b>     | <u>Command</u><br><b>MN</b><br><b>A5</b><br><b>V5</b><br><b>D25000</b><br><b>T2</b><br><b>G</b><br><b>T5</b><br><b>G</b>                 |                   | <u>Description</u><br>Set normal mode<br>Set acceleration to 5 revs/sec <sup>2</sup><br>Set velocity to 5 revs/sec<br>Set distance to 25000 steps<br>Pause for 2 seconds<br>Go<br>5 second time delay<br>After a further pause of 5 seconds, the move is again executed |  |

| <b>TRD</b>         |  | <b>Trigger on Input Distance</b> |  | <b>VALID FROM</b><br>Version 3.0         |
|--------------------|--|----------------------------------|--|--|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>                     | <b>DEFAULT</b>   | <b>ATTRIBUTES</b>                        |
| <a>TRDn            | User steps   | ±2,097,151                       | N/A  | Buffered, Universal, Savable in Sequence |
| <b>Description</b> | Pauses command execution until a specified distance has been reached. It is used in the speed-change mode MQ to cause velocity to change at a specific point, but can also be used for example to turn on an output. The distance specified can be incremental, or absolute in MPA mode. |                                  |  |  |
| <b>Example</b>     | <u>Command</u><br><b>TRD3000</b><br><b>V10</b>   |                                  | <u>Description</u><br>Set speed change distance to 3000 steps<br>Change speed to 10 revs/sec |  |

| <b>TRE</b>            |  | <b>Trigger on Input Equal</b> |                 | <b>VALID FROM</b><br>Version 1.4            |
|-----------------------|--|-------------------------------|-----------------|---|
| <b>SYNTAX</b>         | <b>UNITS</b>   | <b>RANGE</b>                  | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                           |
| <a>TREn               | N/A  | N/A                           | N/A             | Buffered, Universal,<br>Savable in Sequence |
| <b>EXECUTION TIME</b> |  |                               | <b>SEE ALSO</b> | <b>SKE</b>                                  |
| <b>Description</b>    | Pause until input status matches the pattern n.  |                               |                 |   |
|                       | The format for n is bbbbbbb_bbbbbbb, and it is equivalent to:<br><br>HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX<br>TRACK - INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6,<br>INPUT 7.<br><br>Each input can be 0 (off), 1 (on) or X (don't care).<br><br>You can omit trailing X specifiers within each group of 7 specifiers.<br>For example the specifiers:<br><br>001XXXX_XXX11XX and<br>001_XXX11 are equivalent.<br><br>The next two specifiers are also equivalent:<br><br>XXXXXXXX_X1XXXXX and<br>_X1. |                               |                 |   |

| <b>TRIP</b>           |  | <b>Trigger on In Position</b> |                 | <b>VALID FROM</b><br>Version 1.4            |
|-----------------------|--|-------------------------------|-----------------|---|
| <b>SYNTAX</b>         | <b>UNITS</b>   | <b>RANGE</b>                  | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                           |
| <a>TRIP               | N/A  | N/A                           | N/A             | Buffered, Universal,<br>Savable in Sequence |
| <b>EXECUTION TIME</b> |  |                               | <b>SEE ALSO</b> | <b>SSC</b>                                  |
| <b>Description</b>    | After use of this command, further buffered commands are not executed until the motor has stopped within the deadband region for the specified time. The TRIP command can be used in the MC or MQ modes. |                               |                 |   |

**Example**

| <u>Command</u> | <u>Description</u>                            |
|----------------|---|
| <b>MQ</b>      | Set speed change mode                         |
| <b>1SSC1</b>   | Set output 2 as 'in position'                 |
| <b>V35</b>     | Set velocity to 35 revs/sec                   |
| <b>A200</b>    | Set acceleration to 200 revs/sec <sup>2</sup> |
| <b>D80000</b>  | Set distance to 80000 steps                   |
| <b>L</b>       | Set to loop once                              |
| <b>G</b>       | Go - start the move                           |
| <b>TRIP</b>    | Wait in position before repeating move        |
| <b>N</b>       | End of loop                                   |

|            |                                       |                                  |
|------------|---------------------------------------|----------------------------------|
| <b>TRN</b> | <b>Trigger on Input<br/>Not Equal</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|---------------------------------------|----------------------------------|

| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
|---------------|--------------|--------------|----------------|---|
| <a>TRNn       | N/A          | N/A          | N/A            | Buffered, Universal,<br>Savable in Sequence |

**Description**

Pause until the input status is not equal to n.

The format for n is bbbbbbb\_bbbbbbb and it is equivalent to:

HOME, LMT-, LMT+, +JOG, -JOG, EIN (Drive energised), INDEX  
TRACK - INPUT 1, INPUT 2, INPUT 3, INPUT 4, INPUT 5, INPUT 6,  
INPUT 7.

Each input can be 0 (off), 1 (on) or X (don't care).

You can omit trailing X specifiers within each group of 7 specifiers.  
For example the specifiers:

001XXXX\_XXX11XX and  
001\_XXX11 are equivalent.

The next two specifiers are also equivalent:

XXXXXXXX\_X1XXXXX and  
\_X1.

|            |                          |                                  |
|------------|--------------------------|----------------------------------|
| <b>TRR</b> | <b>Registration Mode</b> | <b>VALID FROM</b><br>Version 2.3 |
|------------|--------------------------|----------------------------------|

| SYNTAX  | UNITS      | RANGE                       | DEFAULT | ATTRIBUTES                                  |
|---------|------------|-----------------------------|---------|---|
| <a>TRRn | User steps | -2,097,152 to<br>+2,097,151 | 0       | Buffered, Universal<br>Saveable in sequence |

**Description** Selects Registration move, a move ending a specified distance in user steps after a mark signal appears at Input 6. The move is used in modes MC or MQ. Registration moves are described in detail in Section H.

|             |                                |                                  |
|-------------|--------------------------------|----------------------------------|
| <b>TUNE</b> | <b>Display Tuning Settings</b> | <b>VALID FROM</b><br>Version 2.4 |
|-------------|--------------------------------|----------------------------------|

| SYNTAX   | UNITS | RANGE | DEFAULT | ATTRIBUTES  |
|----------|-------|-------|---------|---|
| aTUNE<n> | N/A   | 1-11  | N/A     | Buffered, Device Specific,<br>Savable in Sequence |

**EXECUTION TIME**

**SEE ALSO** TUNET,  
TUNEV, DR

**Description** The TUNE command with no parameter causes the control module to report tuning information calculated or measured during the previous move. The response format is:

1. Measured largest error - steps.
2. Measured largest torque demand.
3. Measured settling time within window (ms).
4. Measured total indexing + settling time (ms).
5. Calculated stiffness (mNm/step).
6. Calculated error at maximum torque (steps).
7. Estimated load inertia (Kg/cm<sup>2</sup>).
8. Estimated maximum available acceleration (revs/sec<sup>2</sup>).
9. Estimated double time constant (ms).
10. Estimated 1/2 settling time (ms).
11. Estimated closed loop bandwidth (Hz).

If a numeric parameter <n> is entered, any one of the above values as defined by the number will be reported.

If either CPG or CVG is set to zero, the range of values returned will be restricted to the first 4 (i.e. TUNE5 and above are not available).

**Example**

| <u>Command</u> | <u>Description</u>                            |
|----------------|---|
| <b>1TUNET</b>  | Self-tune servo                               |
| <b>CEW2</b>    | Set narrow 'in position' window               |
| <b>D400</b>    | Set distance to 400 steps                     |
| <b>A100</b>    | Set acceleration to 100 revs/sec <sup>2</sup> |
| <b>V50</b>     | Set velocity to 50 revs/sec                   |
| <b>G</b>       | Go - start the move                           |
| <b>1TUNE3</b>  | Report data item 3 (settling time) only       |

|              |   |                                  |
|--------------|---|----------------------------------|
| <b>TUNET</b> | <b>Servo Self-Tuning</b><br>(Torque amplifier settings) | <b>VALID FROM</b><br>Version 2.4 |
|--------------|---|----------------------------------|

|               |              |              |                |   |
|---------------|--------------|--------------|----------------|---|
| <b>SYNTAX</b> | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>TUNET      | N/A          | N/A          | N/A            | Buffered, Universal,<br>Savable in Sequence |

|                       |                                 |
|-----------------------|---------------------------------|
| <b>EXECUTION TIME</b> | <b>SEE ALSO</b> TUNE,<br>TUNEV, |
|-----------------------|---------------------------------|

**Description**

This command self-tunes the servo as a torque amplifier. The following parameters are set:

- CPG    Maximised to the point where the standing digital vibration is reasonable.
- CVG    Set for stability - little overshoot beyond the end point of a move and no oscillations.
- COFF   Set so that the average error is zero. This cancels the effect of amplifier and gravitational offsets on positioning accuracy.
- CTG    Set to cut off the velocity feedback action at high frequency.
- CIG    Set to zero. Integral action is always de-stabilising and a specialised feature associated with a requirement for extreme accuracy.
- CFG    Set to zero, assuming a point to point application. You should set this to the value of CVG (PID equivalent) if you have a following/contouring type application.

**Example**

| <u>Command</u> | <u>Description</u>             |
|----------------|--------------------------------|
| <b>1TUNET</b>  | Self-tune servo                |
| <b>1SV</b>     | Save parameters for future use |

**SAFETY**

The tuning process moves the load a distance of 178 steps backwards and forwards using step changes of position demand that

result in about 50% of the maximum available motor torque being suddenly applied to the load. This process continues for up to 30 seconds depending on the load conditions. During this time buffered commands typed at the keyboard or pending in a sequence will not be executed. However, the unit will respond to immediate commands such as K, S or OFF, to enabled end of travel inputs, to an enabled Input 5 stop input and to the emergency stop input. If the tuning is interrupted in this way, the servo parameters will then be in an intermediate state and it may be necessary to cycle the power or use the Z command to restore them to a viable state.

# TUNEV

## Servo Self-Tuning (Velocity amplifier settings)

**VALID FROM**  
Version 2.4

| SYNTAX   | UNITS | RANGE | DEFAULT | ATTRIBUTES                                |
|----------|-------|-------|---------|---|
| <a>TUNEV | N/A   | N/A   | N/A     | Buffered, Universal, Saveable in sequence |

### EXECUTION TIME

**SEE ALSO** TUNE,  
TUNET,

### Description

This command self-tunes the servo as a velocity amplifier. The process may also be used for pneumatic and hydraulic actuators. It is a limited process, maximising CPG starting from the value you enter so that there is little overshoot and no oscillations occur. The final value of CPG is dependent on the velocity loop gain which you have set. The following parameters are set:

- CPG Maximised, starting from the entered value and taken to the point where there is little overshoot beyond the end of move position and no oscillations occur.
- CVG Not altered - normally zero since damping is provided by the velocity amplifier.
- COFF Set so that the average error is zero. This cancels the effect of amplifier and gravitational offsets on positioning accuracy.
- CTG Not altered.
- CIG Set to zero. Integral action is always de-stabilising and a specialised feature associated with a requirement for extreme accuracy.
- CFG Set to zero.

**Example**

| <u>Command</u> | <u>Description</u>             |
|----------------|--------------------------------|
| <b>1CPG50</b>  | CPG set for a low gain         |
| <b>1TUNEV</b>  | Self-tune servo                |
| <b>1SV</b>     | Save parameters for future use |

**SAFETY**

The tuning process moves the load a distance of 178 steps backwards and forwards using step changes of position demand that result in about 50% of the maximum available motor torque being suddenly applied to the load. This process continues for up to 30 seconds depending on the load conditions. During this time buffered commands typed at the keyboard or pending in a sequence will not be executed. However, the unit will respond to immediate commands such as K, S or OFF, to enabled end of travel inputs, to an enabled Input 5 stop input and to the emergency stop input. If the tuning is interrupted in this way, the servo parameters will then be in an intermediate state and it may be necessary to cycle the power or use the Z command to restore them to a viable state.

**U****Pause****VALID FROM**

Version 1.4

**SYNTAX**

&lt;a&gt;U

**UNITS**

N/A

**RANGE**

N/A

**DEFAULT**

N/A

**ATTRIBUTES**

Immediate, Universal

**EXECUTION TIME****SEE ALSO** PS, C**Description**

Pause immediately and wait for continue.

Execution of a command string will be paused at the point where a U command is entered. Command C will cause execution to be resumed at the point where it was paused.

**Example**

| <u>Command</u> | <u>Description</u>                           |
|----------------|--|
| <b>MN</b>      | Set normal mode                              |
| <b>A10</b>     | Set acceleration to 10 revs/sec <sup>2</sup> |
| <b>V10</b>     | Set velocity to 10 revs/sec                  |
| <b>L</b>       | Loop continuously                            |
| <b>D8000</b>   | Set distance to 8000 steps                   |
| <b>G</b>       | Go - start move                              |
| <b>H-</b>      | Set direction to negative                    |
| <b>G</b>       | Go - start negative move                     |
| <b>N</b>       | End of loop                                  |

In this example the motor will turn 2 revolutions in the positive direction, then two revolutions in the negative direction in a continuous loop. If a U command is entered during the execution of the loop, the motor will stop at the end of its current move. On receipt of a C command, execution of the loop will be resumed from the point where it was paused.

| <b>V</b>           |  | <b>Velocity</b>        |   | <b>VALID FROM</b><br>Version 1.4            |
|--------------------|--|------------------------|---|---|
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>           | <b>DEFAULT</b>  | <b>ATTRIBUTES</b>                           |
| <a>Vn              | Revs/sec   | n = 0.0001 -<br>100.00 | 1   | Buffered, Universal,<br>Savable in Sequence |
|                    |  | <b>EXECUTION TIME</b>  |   | <b>SEE ALSO</b> A, D                        |
| <b>Description</b> | Sets the velocity. Note that the actual top speed of the motor is limited by the motor type and drive. If the velocity is set too high, the following error will exceed the maximum position error (set by CPE) and the drive will be shut down. If the command V0 is given when moving in MQ or MC the motor will stop as in the S command. If the motor is not moving, the fastest possible velocity is used for the move. |                        |   |   |
| <b>Example</b>     | <u>Command</u>   |                        | <u>Description</u>  |   |
|                    | A10  |                        | Set acceleration to 10 revs/sec <sup>2</sup>  |   |
|                    | V10  |                        | Set velocity to 10 revs/sec   |   |
|                    | D8000  |                        | Set distance to 8000 steps  |   |
|                    | G  |                        | Go - Motor will turn 2 revolutions in the positive direction at 10 revolutions per second |   |
| <b>VSY</b>         | <b>Velocity Change -<br/>Synchronous Mode</b>  |                        | <b>Not used in PS7X, BLX</b>  |   |

|                    |   |                       |                |   |
|--------------------|---|-----------------------|----------------|---|
| <b>XC</b>          | <b>Checksum</b>   |                       |                | <b>VALID FROM</b><br>Version 1.4                  |
| <b>SYNTAX</b>      | <b>UNITS</b>  | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                                 |
| aXC                | N/A   | N/A                   | N/A            | Buffered, Device Specific,<br>Savable in Sequence |
|                    |   | <b>EXECUTION TIME</b> |                | <b>SEE ALSO</b> LA                                |
| <b>Description</b> | This command causes the checksums for the volatile and non-volatile RAM to be computed and reported . |                       |                |   |

|                    |  |                       |                |   |
|--------------------|--|-----------------------|----------------|---|
| <b>XD</b>          | <b>Sequence Download</b>   |                       |                | <b>VALID FROM</b><br>Version 1.4            |
| <b>SYNTAX</b>      | <b>UNITS</b>   | <b>RANGE</b>          | <b>DEFAULT</b> | <b>ATTRIBUTES</b>                           |
| <a>XDn             | N/A  | 1 - 63                | N/A            | Buffered, Universal,<br>Savable in Sequence |
|                    |  | <b>EXECUTION TIME</b> |                | <b>SEE ALSO</b> XT, XE, XR,<br>XRP          |
| <b>Description</b> | This is the 'Start sequence' indicator for downloading a sequence. It clears the command buffer and all subsequent commands will be stored in the buffer until receipt of an XT command. A sequence will not be stored if a sequence with the same number identifier already exists, or if there is an error in writing the sequence. THE SEQUENCE IS NOT STORED IN NON-VOLATILE BACKUP RAM UNTIL THE SV COMMAND IS GIVEN. |                       |                |   |

**Example**

| <u>Command</u> | <u>Description</u>                            |
|----------------|---|
| <b>XD1</b>     | Start sequence 1 download                     |
| <b>A10</b>     | Set acceleration to 10 steps/sec <sup>2</sup> |
| <b>V10</b>     | Set velocity to 10 revs/sec                   |
| <b>D8000</b>   | Set distance to 8000 steps                    |
| <b>G</b>       | Go - start move                               |
| <b>D4000</b>   | Set distance to 4000 steps                    |
| <b>H</b>       | Reverse direction                             |
| <b>G</b>       | Go - start move                               |
| <b>XT</b>      | End of download                               |

The sequence is defined as sequence No.1. Each time XR1 is entered the motor will turn 2 revolutions positive and then 1 revolution negative.

**XE****Sequence Delete**

**VALID FROM**  
Version 1.4

**SYNTAX**

&lt;a&gt;XEn

**UNITS**n = sequence  
number**RANGE**

n = 1 - 63

**DEFAULT**

N/A

**ATTRIBUTES**

Buffered, Universal

**EXECUTION TIME****SEE ALSO**XD, XT, XR,  
XRP**Description**

Delete sequence n from RAM.

**Example**

Command  
**XE1 SV**

Description  
Sequence No.1 is deleted from RAM

**XP****Power-On Sequence  
Number**

**VALID FROM**  
Version 1.4

**SYNTAX**

&lt;a&gt;XPn

**UNITS**n = sequence  
number**RANGE**

n = 0 - 63

**DEFAULT**

N/A

**ATTRIBUTES**

Buffered, Universal

**EXECUTION TIME****SEE ALSO**XQ, XSP,  
XSR, XZ, IM, SS**Description**

Set the power-on sequence mode and sequence number to be executed on power up. The command must be saved before power off to enable the sequence to be executed on power up. If the command XP0 is sent, no power-on sequence will be executed. This facility is normally used for stand alone operation.

**Example**

Command  
**XP1**  
**SV**

Description  
 Sequence No.1 will be executed on power-up  
 Save current setup

**XR****Run Sequence**

**VALID FROM**  
 Version 1.4

**SYNTAX**

<a>XRn

**UNITS**

n = sequence  
 number

**RANGE**

n = 1 - 63

**DEFAULT**

N/A

**ATTRIBUTES**

Buffered, Universal,  
 Savable in Sequence

**EXECUTION TIME****SEE ALSO**

XT, XD, XE,  
 XRP

**Description**

Run a sequence.

An XR command can be used within one sequence to start execution of another sequence; all commands in the first sequence following the XR will be executed after the sequence called by the XR command is complete (in this respect an XR acts like a subroutine call).

You must be certain that the number of loops and loop terminators are balanced within a single sequence.

You cannot start a loop in one sequence and end it in another called by the XR command.

**Example**

Command  
**XR1**

Description  
 Sequence No.1 will be executed

**XRP****Run/Pause Sequence**

**VALID FROM**  
 Version 1.4

**SYNTAX**

<a>XRPn

**UNITS**

n = sequence  
 number

**RANGE**

n = 1 - 63

**DEFAULT**

N/A

**ATTRIBUTES**

Buffered, Universal,  
 Savable in Sequence

**EXECUTION TIME****SEE ALSO**

XR, XD, XT,  
 XE

**Description**

Run a sequence with a pause. The XRP command operates in the same way as XR except that a pause condition, which must be cleared before the command buffer is executed, is automatically generated. The pause condition is executed only if the sequence is valid and actually run

**Example**

Command  
**XRP1**

Description  
Sequence No.1 is loaded into the command buffer and is paused until a continue (command C) is entered

|            |                             |                                  |
|------------|-----------------------------|----------------------------------|
| <b>XRT</b> | <b>Return From Sequence</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|-----------------------------|----------------------------------|

|                       |              |              |                    |  |
|-----------------------|--------------|--------------|--------------------|--|
| <b>SYNTAX</b>         | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b>     | <b>ATTRIBUTES</b>                        |
| <a>XRT                | N/A          | N/A          | N/A                | Buffered, Universal, Savable in Sequence |
| <b>EXECUTION TIME</b> |              |              | <b>SEE ALSO</b> XT |  |

**Description**      This command is used in a sequence to terminate execution of that sequence. It is typically used in branched sequences where the RS232C Control Module must exit the sequence before the XT command. It is used in conjunction with 'SKIP' commands.

|            |  |                                  |
|------------|--|----------------------------------|
| <b>XSD</b> | <b>Sequence Download Status Report</b> | <b>VALID FROM</b><br>Version 1.4 |
|------------|--|----------------------------------|

|                       |              |              |                    |                           |
|-----------------------|--------------|--------------|--------------------|---------------------------|
| <b>SYNTAX</b>         | <b>UNITS</b> | <b>RANGE</b> | <b>DEFAULT</b>     | <b>ATTRIBUTES</b>         |
| aXSD                  | N/A          | N/A          | N/A                | Buffered, Device Specific |
| <b>EXECUTION TIME</b> |              |              | <b>SEE ALSO</b> XD |                           |

**Description**      Requests report back of the status of a previous sequence sent from the controller to the RS232C Control Module as a single ASCII character followed by [CR] as follows:

- 0[CR]: Received OK
- 1[CR]: Cannot overwrite existing sequence (sequence with this number already exists - erase previous one first)
- 2[CR]: Sequence buffer full

Note that in order to retain the sequence it is necessary to save it using the SV command.

**Example**

Command  
**XSD**

Response  
0[CR] (Sequence received OK)

| <b>XSR</b>            |  | <b>Sequence Run Status Report</b> |                                    | <b>VALID FROM</b><br>Version 1.4 |
|-----------------------|--|-----------------------------------|------------------------------------|----------------------------------|
| <b>SYNTAX</b>         | <b>UNITS</b>   | <b>RANGE</b>                      | <b>DEFAULT</b>                     | <b>ATTRIBUTES</b>                |
| aXSR                  | N/A  | N/A                               | N/A                                | Immediate, Device Specific       |
| <b>EXECUTION TIME</b> |  |                                   | <b>SEE ALSO</b> XR, XRP            |                                  |
| <b>Description</b>    | Return sequence run status as a single ASCII character followed by [CR].                             |                                   |                                    |                                  |
|                       | 0[CR] = Running (or ran successfully)<br>2[CR] = Invalid sequence - requested sequence doesn't exist |                                   |                                    |                                  |
| <b>Example</b>        | <u>Command</u><br>1XSR   |                                   | <u>Response</u><br>0 (sequence OK) |                                  |

| <b>XSS</b>            |   | <b>X Sequence Status</b> |  | <b>VALID FROM</b><br>Version 1.4 |
|-----------------------|---|--------------------------|--|----------------------------------|
| <b>SYNTAX</b>         | <b>UNITS</b>  | <b>RANGE</b>             | <b>DEFAULT</b>                                   | <b>ATTRIBUTES</b>                |
| aXSSn                 | Sequence number   | n = 1 - 63               | N/A  | Buffered, Device Specific        |
| <b>EXECUTION TIME</b> |   |                          | <b>SEE ALSO</b> XD, XT, XE                       |                                  |
| <b>Description:</b>   | Return status of sequence n as a single ASCII character followed by [CR].   |                          |  |                                  |
|                       | 0[CR]: Empty - sequence doesn't exist<br>2[CR]: OK - sequence does exist  |                          |  |                                  |
|                       | The XSS command will inform the user if the sequence indicated by the number immediately following (no space) the XSS command does in fact exist in memory. |                          |  |                                  |
| <b>Example</b>        | <u>Command</u><br>1XSS1   |                          | <u>Response</u><br>0 [CR] (Sequence not defined) |                                  |

|           |                            |                                  |
|-----------|----------------------------|----------------------------------|
| <b>XT</b> | <b>Sequence Terminator</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|----------------------------|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES                                  |
|--------|-------|-------|---------|---|
| <a>XT  | N/A   | N/A   | N/A     | Buffered, Universal,<br>Savable in Sequence |

**Description** The XT command is used to terminate a sequence download, thus returning the system to command mode. The executable end of sequence is also marked at this point.

|           |                        |                                  |
|-----------|------------------------|----------------------------------|
| <b>XU</b> | <b>Sequence Upload</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|------------------------|----------------------------------|

| SYNTAX | UNITS                  | RANGE      | DEFAULT | ATTRIBUTES                 |
|--------|------------------------|------------|---------|----------------------------|
| aXU<n> | n = sequence<br>number | n = 1 - 63 | N/A     | Immediate, Device Specific |

**Description** The sequence upload command displays the sequences stored in RAM. The value n is the sequence number and if it is omitted a display of all stored sequences will occur. Note the RS232C Control Module is supplied with default sequences to get you started (see RIFS command).

|           |   |                                  |
|-----------|---|----------------------------------|
| <b>XZ</b> | <b>Reset Power-Up<br/>Sequence Mode</b> | <b>VALID FROM</b><br>Version 1.4 |
|-----------|---|----------------------------------|

| SYNTAX | UNITS | RANGE | DEFAULT | ATTRIBUTES           |
|--------|-------|-------|---------|----------------------|
| <a>XZ  | N/A   | N/A   | N/A     | Immediate, Universal |

**EXECUTION TIME** **SEE ALSO** Z, XP

**Description** Sets the power-on sequence mode to zero (thereby disabling sequence activation on power-on).

This is the same function as XP0 but it also calls the SV command to save this mode.



## Command summary

### Help Command (HELP1)

|        |                             |
|--------|-----------------------------|
| HELP   | Produce all help displays   |
| HELP1  | HELP list                   |
| HELP2  | Indexer Commands            |
| HELP3  | Start/Stop Commands         |
| HELP4  | Sequence Commands           |
| HELP5  | Homing, Jog, Limit          |
| HELP6  | Option Set Commands         |
| HELP7  | Information Commands        |
| HELP8  | Inputs and Outputs          |
| HELP9  | Control and Memory Commands |
| HELP10 | Using the Indexer           |
| HELP11 | Recommended Servo Settings  |
| HELP12 | Servo Setup Procedure       |

### Motion Commands (HELP2)

|      |                              |
|------|------------------------------|
| A(n) | Set acceleration             |
| C    | Continue                     |
| D(n) | Set move distance            |
| FDA  | Set feed acceleration        |
| FDV  | Set feed velocity            |
| G    | Go, make a move              |
| GL   | Go linear                    |
| GLD  | Go linear define             |
| GSY  | Go synchronous               |
| H    | Reverse motor direction      |
| H+   | Set positive direction       |
| H-   | Set negative direction       |
| ID   | Immediate distance           |
| IV   | Immediate velocity           |
| L(n) | Loop instructions            |
| MAS  | Mode asynchronous            |
| MC   | Move continuous              |
| MN   | Normal preset moves          |
| MPA  | Absolute position mode       |
| MPI  | Incremental position mode    |
| MQ   | Velocity change preset moves |
| MSY  | Mode synchronous             |
| N    | End of loop indicator        |
| PS   | Wait for continue            |
| PSY  | Pause synchronous            |
| SIM  | Set indexer/following mode   |
| T(n) | Delay for n secs             |
| TRD  | Pause until distance reached |
| TRE  | Pause until inputs equal     |
| TRIP | Pause until in position      |

|      |                              |
|------|------------------------------|
| TRN  | Pause until inputs not equal |
| TRR  | Pause for I6 registration    |
| U    | Pause, wait for continue     |
| V(n) | Set velocity                 |
| VSY  | Synchronise velocity change  |

**Start/Stop Commands  
(HELP3)**

|     |                               |
|-----|-------------------------------|
| K   | Kill motion                   |
| LS  | Stop now at LA deceleration   |
| OFF | Immediate stop - de-energise  |
| ON  | Immediate start - re-energise |
| S   | Stop motion, decelerate now   |
| SB  | Stop motion, buffered         |
| ST0 | Energise drive                |
| ST1 | De-energise drive             |
| Y   | Terminate loop or sequence    |
| Z   | Power up reset                |

**Sequence Commands  
(HELP4)**

|        |                                 |
|--------|---------------------------------|
| XC     | Report checksum                 |
| XD(n)  | Begin download sequence         |
| XE(n)  | Delete sequence                 |
| XP(n)  | Select power up sequence        |
| XR(n)  | Execute sequence                |
| XRP(n) | Execute sequence with pause     |
| XRT    | Return from sequence            |
| XSD    | Report status of download       |
| XSR    | Report run sequence status      |
| XSS(n) | Report sequence status          |
| XT     | Sequence terminator             |
| XU(n)  | Sequence upload                 |
| XZ     | Turn off power up sequence mode |

**Homing, Jog, Limit  
Commands (HELP5)**

|           |                                       |
|-----------|---------------------------------------|
| GA(n)     | Set go home acceleration              |
| GH+/- (n) | Go home with direction + velocity     |
| GHF(n)    | Final home move velocity              |
| JA(n)     | Set jog acceleration                  |
| JV(n)     | Set jog velocity                      |
| LA(n)     | Set limit accn                        |
| LD(n)     | Set limit function                    |
| PZ        | Set current position as absolute zero |
| SP        | Set current position to value         |

**Option Set Commands  
(HELP6)**

|     |  |
|-----|--|
| E   | Enable communications  |
| EX  | 1=terminal 0=computer mode   |
| F   | Disable communications   |
| HS  | Display H/W switches   |
| HSA | 1=resolver feedback (not available)  |
| HSB | 0=closed loop brushless  |
| HSC | 0=open loop stepper 1=other  |
| HSD | 0=no daughter board 1=output expansion board fitted 2=analogue feedback board fitted |
| OS  | Display 'OS' switches  |
| OSA | 1=Home to index pulse  |
| OSB | 1=switched integral action on  |
| OSC | 1=single value returned by DS  |
| OSD | 1=input 7 is sequence select line  |
| OSE | 1=enable jog   |
| OSF | 1=allow initialise on limit  |
| OSG | 1=allow dither   |
| OSH | 1=square dither 0=triangular   |
| OSI | 1=reduced movement initialise  |
| OSJ | 1=24 bit gearbox 0=16 bit  |
| OSK | 1=encoder count checked  |
| OSM | 1=accumulate at 2ms sampling   |
| OSO | 1=suppress units   |
| SS  | Display SS switches  |
| SSA | 0=echo on 1=echo off   |
| SSB | 1=input 6 clears and store pause   |
| SSC | 1=output 2 is in position  |
| SSD | 1=output 1 is composite fault  |
| SSE | 1=input 5 is stop line   |
| SSF | 1=input 4 is sequence select   |
| SSG | 1=save comm buffer on limit  |
| SSH | 1=save command buffer on stop  |
| SSI | 1=inputs 5,6 & 7 select sequences  |

**Information Commands  
(HELP7)**

|     |                                |
|-----|--------------------------------|
| B   | Command buffer status request  |
| BS  | Command buffer size request    |
| DFX | Returns index status (32 bits) |
| DIC | Report Control Module position |
| DPA | Display position-actual        |
| DPE | Continuous error display       |
| DPS | Display position setpoint      |
| DR  | Display report                 |
| DVA | Display velocity-actual        |
| DVS | Display velocity setpoint      |
| OSC | 1=report one signal value only |

|     |                               |
|-----|-------------------------------|
| P   | Request position this move    |
| PIC | Display servo loop diagram    |
| PR  | Request absolute position     |
| QS  | Send a response at move end   |
| R   | Status request                |
| RA  | Limit switch status request   |
| RB  | Pause status request          |
| RE  | Report drive status           |
| RG  | Go home status request        |
| RPO | Report power on time          |
| RS  | Sequence status request       |
| RSE | Report servo error conditions |
| RV  | Report software revision      |

#### Inputs and Outputs (HELP8)

|     |   |
|-----|---|
| HSD | 0=no daughter board 1=output expansion board fitted 2=analogue board fitted |
| IO  | Sets outputs immediately  |
| IS  | Reports input status  |
| O   | Turns outputs on and off  |
| OSD | 1=input 7 is sequence select line   |
| SKE | Skip on inputs equal  |
| SKN | Skip on inputs not equal  |
| SSB | 1=input 6 clears pause  |
| SSC | 1=output 2 is in position   |
| SSD | 1=output 1 is composite fault   |
| SSE | 1=input 5 is controlled stop line   |
| SSF | 1=input 4 is sequence select strobe   |
| SSG | 1=save command buffer on limit  |
| SSH | 1=save command buffer on stop   |
| SSI | 1=inputs 5,6,7 are sequence select  |
| TRE | Pause until inputs equal  |
| TRN | Pause until inputs not equal  |

#### Control and Memory Commands (HELP9)

|      |   |
|------|---|
| CAG  | Configure acceleration gain                     |
| CCP  | Configure command peak                          |
| CCS  | Configure command source                        |
| CDG  | Set D gain in PID                               |
| CEW  | Set in position window                          |
| CFG  | Set feed forward gain                           |
| CIG  | Set integral action gain                        |
| CIT  | In position wait time (2ms ints)                |
| CIW  | Set integral action window                      |
| CJL  | Set motor + load inertia (Kg-cms <sup>2</sup> ) |
| CMR  | Set motor resolution                            |
| COFF | Set amplifier offset                            |
| CPE  | Set position error limit                        |

|       |                                     |
|-------|-------------------------------------|
| CPG   | Set forward path gain               |
| CTG   | Set filter time constant            |
| CTQ   | Set available motor torque (Nm)     |
| CUR   | Set user resolution                 |
| CVG   | Set velocity feedback gain          |
| DS    | Signal display by number            |
| DTA   | Set dither amplitude                |
| DTF   | Set dither frequency                |
| FOL   | Set following percent               |
| OFF   | Immediate de-energise               |
| ON    | Immediate energise                  |
| RAT   | Set rate multiplier value           |
| RFS   | Servo to factory settings           |
| RIFS  | Indexer to factory settings         |
| RSE   | Reports servo error conditions      |
| SIM   | Select indexer operation            |
| ST0   | Energise drive                      |
| ST1   | De-energise drive                   |
| SV    | Save current settings in backup RAM |
| TUNE  | Report tuning information           |
| TUNET | Automatic torque amp tuning         |
| TUNEV | Automatic velocity amp tuning       |

**Following Commands  
(HELP13)**

|      |                                 |
|------|---------------------------------|
| CAG  | Set following acceleration gain |
| CCS0 | x4 following decode             |
| CCS1 | x2 following decode             |
| CCS2 | x1 following decode             |
| CCS3 | Clock and Direction Decode      |
| CUR  | Set user resolution             |
| CMR  | Set feedback (motor) resolution |
| FOL  | Set following (SIM5 mode)       |
| RAT  | Set gearbox scaling factor      |
| SIM0 | Indexer, not following, mode    |
| SIM1 | Encoder following mode          |
| SIM2 | Superposition following mode    |
| SIM3 | Positive scaled following       |
| SIM4 | Negative scaled following       |
| SIM5 | Preset following indexer mode   |

## ALPHABETICAL COMMAND LISTING

|      |   |      |   |
|------|---|------|---|
| A    | Acceleration Rate                           | GL   | Go Linear   |
| B    | Buffer Status Request                       | GLD  | Go Linear Define                                    |
| BS   | Buffer Size Request                         | GSY  | Go Synchronous                                      |
| C    | Continue                                    | H    | Change Direction                                    |
| CAG  | Configure Acceleration Gain                 | ^H   | Backspace   |
| CCP  | Configure Command Peak                      | H+   | Set Direction                                       |
| CCS  | Configure Command Source                    | H-   | Set Direction                                       |
| CEW  | Configure In-Position Window                | HELP | Produce Help Screens                                |
| CDG  | Configure Derivative Gain                   | HS   | Display H/W Configured<br>Switches                  |
| CFG  | Configure Feedforward Gain                  | HSB  | Configure Motor Type                                |
| CIG  | Configure Integral Gain                     | HSC  | Configure Motor Type                                |
| CIT  | Configure In-Position Time                  | HSD  | Daughter Board Select                               |
| CIW  | Configure Integral Action<br>Window         | ID   | Immediate Distance                                  |
| CIX  | Configure Index Mark<br>Resolution          | IO   | Immediate Output                                    |
| CJL  | Enter Motor + Load Inertia                  | IS   | Input Status  |
| CMR  | Configure Motor Resolution                  | IV   | Immediate Velocity                                  |
| COFF | Configure Amplifier Offset                  | JA   | Jog Acceleration                                    |
| CPE  | Configure Position Error                    | JV   | Jog Velocity  |
| CPG  | Configure Proportional Gain                 | K    | Kill  |
| CTG  | Configure Filter Time Constant              | KILL | Kill  |
| CTQ  | Enter Motor Torque                          | L    | Loop  |
| CUR  | Configure User Resolution                   | LA   | Limit Deceleration                                  |
| CVG  | Configure Velocity Gain                     | LD   | Limit Disable                                       |
| D    | Distance                                    | LS   | Limit Switch Fast Stop                              |
| DFX  | Display Flags Indexer                       | MAS  | Mode Asynchronous                                   |
| DIC  | Display Indexer Counter                     | MC   | Mode Continuous                                     |
| DPA  | Display Position Actual                     | MN   | Mode Normal   |
| DPE  | Display Position Error                      | MPA  | Mode Position Absolute                              |
| DPS  | Display Position Setpoint                   | MPI  | Mode Position Incremental                           |
| DR   | Display Report                              | MQ   | Speed Change Mode                                   |
| DS   | Display Signal                              | MSY  | Mode Synchronous                                    |
| DTA  | Set Dither Amplitude                        | N    | End Loop  |
| DTF  | Set Dither Frequency                        | O    | Programmable Output                                 |
| DVA  | Display Velocity Actual                     | OFF  | De-Energise Drive                                   |
| DVS  | Display Velocity Setpoint                   | ON   | Energise Drive                                      |
| E    | Enable Communications                       | OS   | Other Switches                                      |
| EX   | Set Communication Style                     | OSA  | Home to an Index Pulse                              |
| F    | Disable Communications                      | OSB  | Integral Action Selection                           |
| FDA  | Acceleration, Linear<br>Interpolation Moves | OSC  | Monitor Command Reporting                           |
| FDV  | Velocity, Linear interpolation<br>Moves     | OSD  | Input 7 Sequence Select                             |
| FOL  | Following Percent                           | OSE  | Jog Enable  |
| G    | Go  | OSF  | Initialisation on Limit                             |
| GA   | Go Home Acceleration                        | OSG  | Dither Enable/Disable                               |
| GH   | Go Home                                     | OSH  | Dither Square/Triangular Select                     |
| GHF  | Go Home Final                               | OSI  | Select Standard/Reduced<br>Movement on Initialising |
|      |   | OSJ  | RAT 16/24 Bit select                                |
|      |   | OSK  | Integrity Check of Encoder                          |

|      |   |       |                                      |
|------|---|-------|--------------------------------------|
| OSM  | Integral Action Sensitivity             | TRN   | Trigger On Input Not Equal           |
| OSO  | Suppress Units                          | TRR   | Registration Mode                    |
| P    | Position                                | TUNE  | Show Tuning Settings                 |
| PIC  | Picture                                 | TUNET | Self-Tune Servo (Torque Amplifier)   |
| PR   | Position Report                         | TUNEV | Self-Tune Servo (Velocity Amplifier) |
| PS   | Pause                                   | U     | Pause                                |
| PSY  | Pause for Synchronisation               | V     | Velocity                             |
| PZ   | Position Zero                           | VSY   | Change velocity, Synchronous Mode    |
| QS   | Transmit An Identifier                  | XC    | Checksum                             |
| R    | Report Control Module Status            | XD    | Sequence Download                    |
| RA   | Report A - Limit Status Request         | XE    | Sequence Delete                      |
| RAT  | Set Rate Multiplier Value               | XP    | Power-On Sequence Number             |
| RB   | Report B - Miscellaneous Status Request | XR    | Run Sequence                         |
| RE   | Drive Status Request                    | XRP   | Run/Pause Sequence                   |
| RFS  | Return Servo to Factory Settings        | XRT   | Return From Sequence                 |
| RG   | Report Go Home Status                   | XSD   | Sequence Download Status Report      |
| RIFS | Return Indexer to Factory Settings      | XSR   | Sequence Run Status Report           |
| RPO  | Report Power-On Time                    | XSS   | X Sequence Status                    |
| RS   | Report Sequence Status                  | XT    | Sequence Terminator                  |
| RSE  | Report Servo Errors                     | XU    | Sequence Upload                      |
| RV   | Revision                                | XZ    | Reset Power-Up Sequence Mode         |
| S    | Stop                                    | Y     | Terminate Loop                       |
| SAVE | Save                                    | Z     | Reset                                |
| SB   | Stop Buffered                           |       |                                      |
| SIM  | Set Indexer/Following Mode              |       |                                      |
| SKE  | Skip On 'Equals'                        |       |                                      |
| SKN  | Skip On 'Not Equal'                     |       |                                      |
| SP   | Set current position to value           |       |                                      |
| SS   | Set Switches                            |       |                                      |
| SSA  | RS232C Echo Control                     |       |                                      |
| SSB  | Set Input 6 = Clear Pause Input         |       |                                      |
| SSC  | Set Output 2 = In-Position Output       |       |                                      |
| SSD  | Set Output 1 as Composite Fault Signal  |       |                                      |
| SSE  | Set Input 5 = Controlled Stop Input     |       |                                      |
| SSF  | Set Input 4 as Sequence Strobe          |       |                                      |
| SSG  | Save Command Buffer On Limit            |       |                                      |
| SSH  | Save Command Buffer On Stop             |       |                                      |
| SSI  | All Inputs Select Sequence              |       |                                      |
| ST   | Energise/De-Energise Drive              |       |                                      |
| STOP | Stop                                    |       |                                      |
| SV   | Save                                    |       |                                      |
| T    | Time Delay                              |       |                                      |
| TRD  | Trigger On Input Distance               |       |                                      |
| TRE  | Trigger On Input Equal                  |       |                                      |
| TRIP | Trigger On In Position                  |       |                                      |

---

## Chapter 6. SERVO TUNING

---

### Tuning the Drive with a Positioner

The positioner is designed to operate with the drive configured as a torque amp, and this is the usual mode of operation described here. In a few applications, it is desirable to configure the drive as a velocity amp, in which case the drive Time Constant and Damping pots are set up as described in the drive manual and the only gain which needs to be set on the positioner is CPG (small values of CVG and CFG can be tried, if desired).

---

### Tuning Positioner with Drive in Torque Amp

To obtain smooth rotation at optimum speed and accurate positioning of the motor without overshoot or oscillation, with the motor connected to your system, you must tune the gain and damping characteristics of the servo loop. The TUNET command provides a self-tuning feature but for some applications the best results will be obtained by manual tuning using the commands listed at the end of this appendix, together with those from the Detailed Command List in Chapter 5. Refer to the servo loop block diagram shown in Figure 1-12. The parameters affecting tuning are as follows.

---

#### Tuning Parameters

##### Proportional gain (CPG command)

Proportional gain determines the amount of torque produced in response to a given position error. It sets the stiffness of the system and affects the following error. A high proportional gain gives a stiff, responsive system but results in overshoot and oscillation which require damping.

##### Velocity gain (CVG command)

Velocity feedback is a signal which increases with shaft speed. It acts in a negative sense opposing the proportional action and helping to stabilise the motion. The damping action of velocity feedback allows a higher proportional gain to be used.

##### Integral gain (CIG command)

Proportional action may be insufficient to overcome static position errors due to friction. Integral action accumulates a steady state error until sufficient torque is produced to move the load. It improves overall positioning accuracy but may produce low frequency oscillation around the commanded position.

##### Feedforward gain (CFG command)

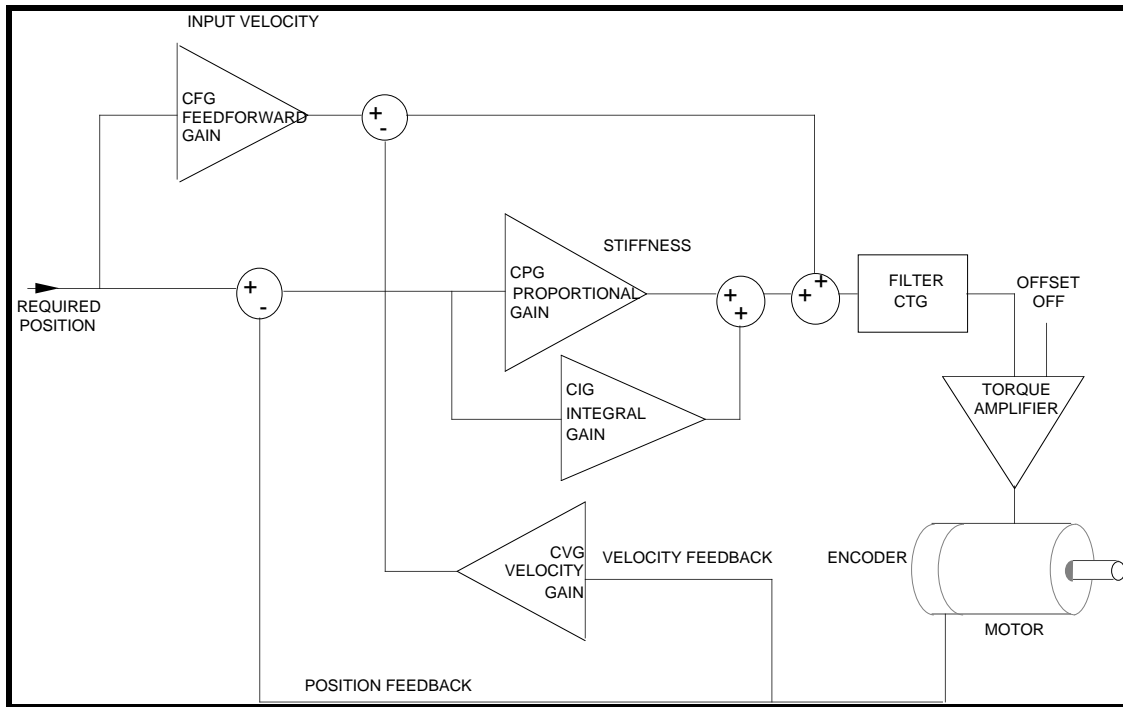
The opposing action of proportional and velocity gains result in a position error which depends on speed. This is called "following error". Feedforward gain can be used to offset the following error and improve tracking accuracy. This is important in contouring applications.

##### Filter time constant (CTG command)

Fast positioning systems need high proportional and velocity gains. By limiting the bandwidth, the digital filter prevents a high gain system from becoming too lively. The filter also serves to average the effects

of the digital control loop, reducing the jitter at standstill and the audible noise. The value of CTG should be kept as low as possible.

You can examine these parameters by using the DR command.



**Figure 6-1. Servo Control Loop**

---

**Terminal/  
Computer**

The terminal/computer communication should be set to to 9600 baud, no parity, full duplex (not local), 1 stop bit.

After power up, the sign on message "DON'T PANIC Type 'HELP'" and the > prompt will appear on the screen unless you have previously set the positioner into the terse computer communications mode using EX0. In any case using the command 1DR will display the status of the positioner.

---

**Connecting the Loop**

**Safety**

It is advisable to have the ESTOP input wired to a stop switch located in an easily accessible position so that the system can be halted quickly in the event of instability causing oscillation. An alternative is to cut the power using the command OFF[CR] but this takes longer.

Unless you have saved data in the past while the motor was energised, the positioner will power on without attempting to move or energise the motor.

If any of the motor or encoder connections have been re-made or changed, the first step is to check the integrity and polarity of the motor connections. Proceed as follows:-

Make sure the axis is de-energized, type OFF[CR], if in doubt. Use the command :

aDS3[CR]

where a = the device address  
to display the feedback position continuously (approximately 0).

Turn the motor shaft CW and check that the displayed value increases, turn the shaft CCW and it should decrease. If this does not happen you must not close the loop with ON at any time since the feedback will be positive.

Press [CR] to restore the > prompt.

---

### **Manual Tuning Procedure**

#### **1. Initial Setting Up**

Type the commands CFG0 CPG0 CVG0 COFF0 CIG0 CTG0 to software disable the servo. Check the result of these commands with command 1DR.

When the axis is energised, it will be open loop with a slight torque demand due to amplifier offsets. This may cause motion of the motor. You may have to restrain the axis and be prepared to switch off, or type OFF[CR]. In any case there is an automatic software limit set by the command CPE on the distance it will drift.

#### **2. Position Error Window Setting**

The drive is disabled to protect the system if the position error goes outside of the position error window. The window can be set to a low number of steps to avoid damage due to overshoot or oscillation.

The value of CPE defines the size of the window in steps. It can be displayed by using the CPE command.

During tuning, CPE is normally set to 4000 steps i.e. 1 rev either way at the usual 4000 steps/rev. If in doubt set a lower value. The effect of going beyond the set error limit will be to automatically de-energise the motor.

Type the command RSE (report servo errors). This may show quite a list of errors. "Drive disabled by software command", "error limit exceeded" and "failed checksum of backup ram" are all acceptable at this point but "Drive fault" should not come up. If it does, check the LED on the drive front panel and try powering down the system and then powering it up again.

### **DO NOT PROCEED IF A DRIVE FAULT PERSISTS**

- |                                    |  |
|------------------------------------|--|
| <b>3. Energising the Axis</b>      | Energise the axis by typing ON [CR]. You may find it convenient to restrain the axis to stop it drifting out of the error limit and de-energising. If it does de-energise, it can be restarted by typing ON.   |
| <b>4. Cancelling Drift</b>         | <p>Cancel any offset using the command aCOFF&lt;value&gt;[CR]. Negative values stop CW drift; positive values stop CCW drift. Values up to a few thousand are acceptable (the maximum is 32,000 corresponding to full torque demand). Only small values are normally required, for example 1COFF10.</p> <p>Check that making COFF more positive tends to turn the motor CW and making it more negative tends to turn it CCW. If this does not happen, do not proceed, as the loop will have a net positive feedback and will oscillate (although probably slowly) when the loop is closed.</p>   |
| <b>5. Closing the Loop</b>         | <p>Set a small position error limit (2000 steps for example) so that if oscillation occurs during tuning, the resulting overshoot producing large position errors will exceed the limit set by CPE and shut down the drive.</p> <p>If necessary first move the axis to a good starting point, then type ON[CR] to energise the motor and CPG1[CR] to close the loop. If a slight vibration appears at this point, it indicates positive feedback. This should only occur if you have changed the ,motor or encoder connections. Taking care, displace the axis slightly and then let go. You should experience a slight spring effect and the axis should return to its original position approximately. The closed loop is now working.</p> |
| <hr/>                              |  |
| <b>Setting Up Servo Parameters</b> | <b>NOTE:</b> When setting up servo parameters, do not allow the motor to go unstable for more than a second or two.  |
| <b>Tracking Applications</b>       | For tracking applications (PID tuning), where minimum following error is required, velocity feedforward should be used to improve the tracking accuracy. This can be set using the CFG command or the CDG command can be used to set feedforward gain and velocity feedback gain to the same value.  |
| <b>Point to Point</b>              | For fast point to point positioning applications (PIVF tuning), a high   |

**Positioning Applications**

proportional gain (CPG) damped by velocity gain (CVG) should be used.

**Help 11**

A guide to servo loop parameter settings for a fully damped non-oscillating response is provided in HELP 11. This uses the maximum low speed motor torque entered by the CTQ command and the total moment of inertia entered by the CJL command in calculating suggested settings for the loop parameters.

First enter the values of CTQ and CJL (Motor + load inertia) for your system, then use HELP 11 to display recommended servo loop parameter settings calculated from the data you have entered. For example, if 3.6 is entered for the CTQ command to indicate a maximum torque of 3.6 Nm available and 1.6 is entered for the CJL command indicating a total inertia of 1.6 Kg/cm<sup>2</sup>, the following table will be displayed:

| Gain CPG | Velocity Gain<br>CVG/CFG/<br>CDG | Stiffness<br>mNm/<br>step | Error @<br>maximum<br>torque (fb<br>steps) | Doubled T<br>Constant<br>(msecs) | 1/2 moved<br>time<br>(msecs) | Closed loop<br>bandwidth<br>(3db) (Hz) |
|----------|----------------------------------|---------------------------|--|----------------------------------|------------------------------|--|
| 92       | 57                               | 10                        | 360  | 5                                | 8                            | 20                                     |
| 46       | 40                               | 5                         | 720  | 7                                | 12                           | 14                                     |
| 23       | 29                               | 2                         | 1400                                       | 10                               | 17                           | 10                                     |
| 11       | 20                               | 1                         | 2900                                       | 14                               | 24                           | 7                                      |
| 5        | 14                               | 0                         | 5700                                       | 20                               | 34                           | 5                                      |

**Table 6-1. HELP 11 - Servo Setup Table**

Each successive line of the table corresponds to a less highly tuned servo system with reduced stiffness and speed of response. These reduced values may be used in matching the servo to lower specification mechanics.

The following describes in more detail the terms used in the table:

**Velocity Gain**

The settings of CVG, CDG and CFG depend upon the application as described above.

For the figures in the rest of the line of the table to apply, the CDG or CVG setting should be as in column 2.

For PID applications, setting CDG to the figure quoted in the table will automatically set CVG and CFG to the same value.

**Stiffness**

Stiffness is the restoring force per unit of displacement. The force is dependent on the number of steps displaced from the demanded position. For example, a displacement of 100 steps with a stiffness of 10 mNm/step would produce a restoring force of 1 Nm.

**Error at Maximum**

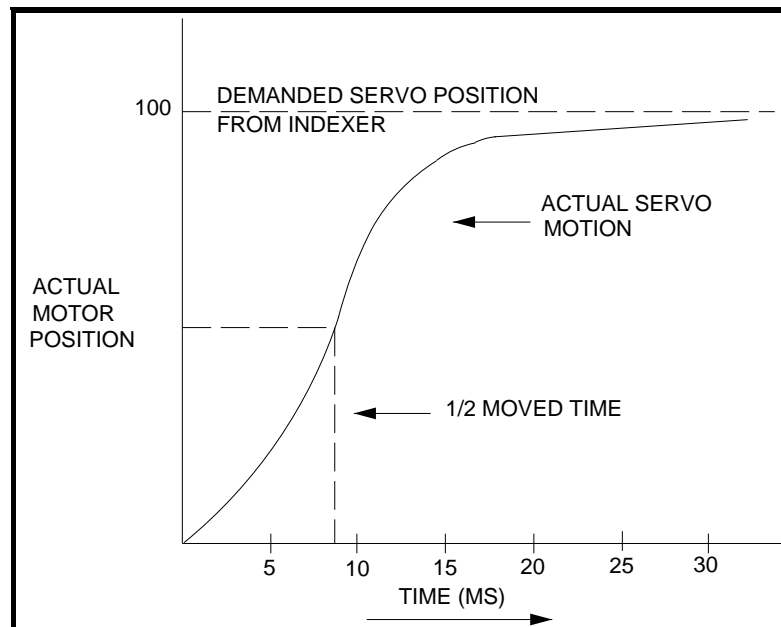
As the axis is increasingly displaced from the demanded position,

**Torque** the restoring torque increases at the quoted stiffness until the maximum available torque is applied. The Error @ Max Torque is the number of steps of displacement required to produce a restoring force of the maximum available torque.

**Doubled T Constant** When the damping is critical, the response of the servo loop is that of two simple filters in series, each having a time constant T. The value of T is quoted in the table.

**1/2 Moved Time** The following is an example move using servo values taken from the top line of the HELP 11 table to show the 1/2 moved time. The move is at maximum acceleration and velocity and it is initiated by the command string: D100 A0 V0 G

Figure 6-2 represents instantaneous position plotted against time for the resulting servo movement .



**Figure 6-2. Example Servo Response Curve**

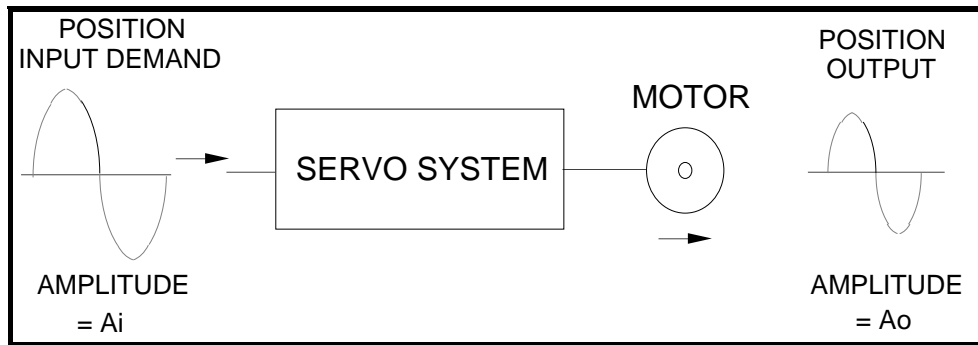
The 1/2 moved time is the time taken to complete half of the move, 50 steps in this example.

The move will be half complete in the time  $1.7T$  (column 6 of the table) or  $1.7 \times 5 = 8$  milliseconds approx.

The move will be 96% complete at time  $5T$ . In the example, the motor will have moved 96 steps in 25ms.

**Bandwidth** The bandwidth of the servo system is the range of frequencies at which a sine wave move pattern input will be followed within 3db by the actual motion of the motor. Figure 6-3 shows the sine wave position demand input and the graphical representation of the motor

movement.



**Figure 6-3. Servo System Bandwidth**

As the frequency of the sine wave of demanded positions at the input ( $A_i$ ) increases, the ability of the servo system to follow is reduced and the amplitude of the sine wave of the resulting output positions ( $A_o$ ) decreases. The frequency at which it is reduced by 3db is the upper limit of the bandwidth.

### Coarse Tuning the Loop

Attach a nylon cable tie to the shaft of the motor. The loose end can be used as a coarse indicator for overshoot and oscillation.

Use the command string `A0 V0 D200 G` to produce a short, sharp rotation of the shaft. Observe the movement, watching the cable tie for overshoot and oscillation. Repeat the movement, increasing CPG using the settings suggested in HELP11 until overshoot just occurs.

Balance the increasing gain with damping by setting CVG to 10, 20, 40 ... Increase it until the oscillation and overshoot indicated by the cable tie ceases.

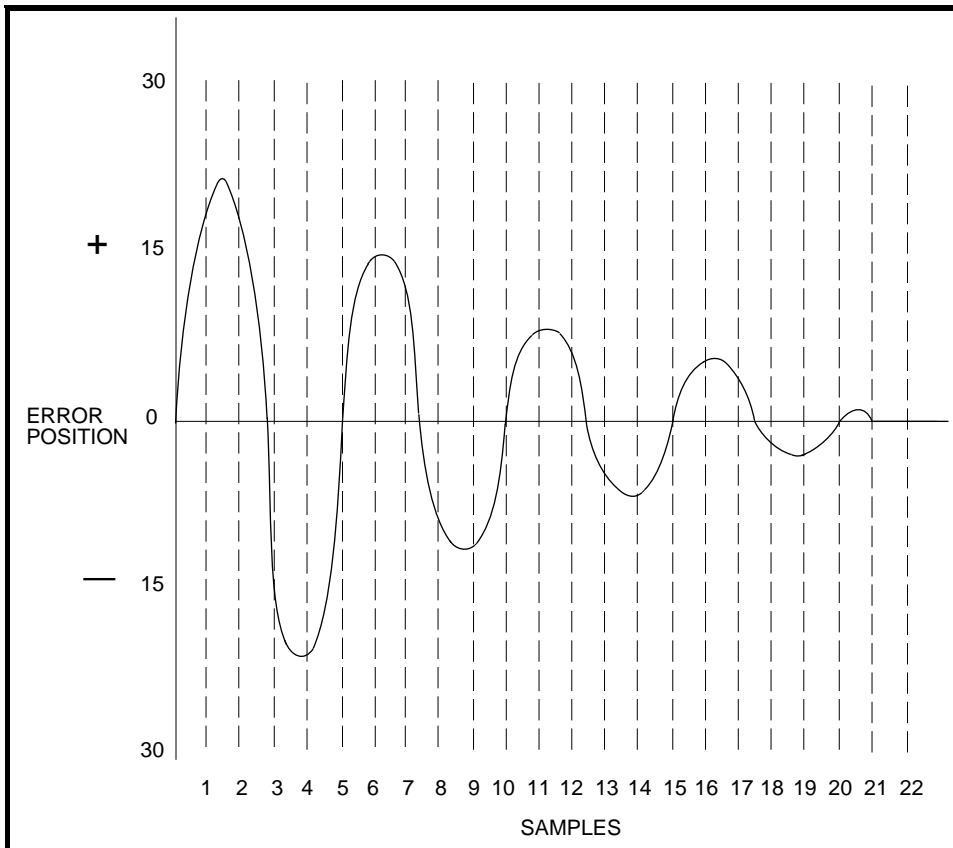
### Fine Tuning

After coarse tuning the loop continue balancing the gain and damping, monitoring the overshoot at the terminal by typing:

```
aDPE [space] G [space] [CR]
```

where a=the device address.

The CR should be entered quickly after typing the space following the G command - this will stop the position error display with the final positioning readings on screen to allow you to analyse them. Overshoot is indicated by the position error reading passing through zero at the end of the movement, then returning back through zero in the opposite direction, oscillating with decreasing error readings about the zero position before finally settling within the position error window. The damping should be increased to the point where overshoot just disappears. Too much damping will result in a sluggish movement.



**Figure 6-4. Overshoot During Positioning**

The following table shows the DPE readings for the final positioning graph of Figure 6-4. This is only an example to illustrate the general principles involved in tuning - in practice, the resulting figures may differ considerably from those given but they will follow the general pattern shown. The overshoot shown here is rather more severe than might be generally expected.

| Samples | Readings |     |     |     |    |    |    |
|---------|----------|-----|-----|-----|----|----|----|
| 1 - 7   | 23       | 22  | -17 | -20 | -5 | 14 | 12 |
| 8 - 14  | -7       | -10 | 0   | 6   | 4  | -3 | -4 |
| 15 - 21 | -2       | 4   | 3   | -2  | -3 | -1 | 0  |

**Notes on  
Tuning****Table 6-2. DPE Readings During Final Positioning**

The TUNE command will show how much torque was used, with 1023 corresponding to saturation. If the maximum torque demand is significantly less than 1023 you should try increasing CPG. You will normally need to increase CVG as well in order to prevent overshoot. Make sure that the maximum torque never exceeds about 1000 to guarantee that the servo loop does not saturate.

The TUNE command also shows the time taken for the complete move as well as the settling time. This information is also useful in optimising the response.

To summarise:

1. Use DPE to look for overshoot and oscillation.
2. Use TUNE to check maximum torque demand, move time and settling time.

CFG, the feedforward gain should never be set very much larger than the velocity feedback gain (CVG). A setting of 20% more than CVG is the largest practical value.

An increase in gain is always best matched by an increase in damping; when this is no longer true you have reached the optimum values. A useful guideline is that an increase of times X in gain is best matched by an increase of times square root X in velocity feedback (think of this as an X% increase in gain matches an X/2% increase in damping e.g. 30% more gain matches 15% more damping). This will keep the shape of the response the same and just increase its frequency (you are holding a quantity called the damping ratio constant).

---

**Servo Self-  
tuning**

A facility for self-tuning the servo is included in the positioner software, using the command TUNET. The servo parameters are set at the time that they are executed and there is no continuous on-going adjustment (known as adaptive control).

Tuning is for the following characteristics:

- Stability - does not oscillate as it moves or settles
- Maximised stiffness and accuracy - deflects as little as possible in response to external forces
- Maximised bandwidth and speed of response - settles to the commanded position as quickly as possible at the end of a move
- Does not overshoot the end position in a move
- Minimal sound during motion or when holding position

- Tracks a profile of constantly changing position or velocity with as little error as possible
- Resistant to change as friction and other load characteristics vary from time to time, machine to machine, application to application and with production variations
- Operates for smaller loads than that used for tuning down to no load.

There are a number of other move related requirements:

- Make full use of the torque and power available so that the longest, fastest move possible can be made
- Perform equally well for fast or slow motions and for short or long moves
- Minimise heating effects in the motor and drive

It is not correct to adjust the servo loop parameters to meet these requirements as the general servo performance would be adversely affected. They are best met by adjusting the move profile (D, V and A values) and possibly the feedforward CFG value whilst watching for saturation effects.

If the application is purely velocity control with no requirement for positioning, then the tuning is best carried out manually. The self-tuning settings will work, but they will not produce the best achievable results.

---

# Index

---

- "\$" prompt 51
- + and - Limits 8
- +Jog, -Jog 8
- Absolute position 104
- Absolute zero 105, 120
- Address jumper links 33
- Address setting 31
- AN-OUT 1 9
- AN-OUT2 10
- Assumptions v
  
- Bandwidth 154
- Basic electronics v
- Basic motion control v
- Buffer Capacity 39
- Buffered 57
- Buffered Clock Mode 28
- Buffered Commands 39
  
- Checksum 50, 134
- Clock signal 7
- Coarse Tuning 154
- Command buffer 40, 50, 58, 59, 123, 124
- Communications protocol 31
- Complex Velocity Profile Using MQ Mode 95
- Composite Fault Indicator 17, 122
- Computer host mode 37
- Contents of This Manual v
  
- Damping 154
- Default values 31
- Developing Your Application v
- Device Specific 57
- Direction 7
- Direction inputs + and - 7
- Disables command execution 77
- Doubled T Constant 153
  
- E stop 8
- Echoback 54
- Encoder following 25
- Encoder Input 7
- Error at Maximum Torque 153
- ESTOP input 149
  
- Feedforward gain 148, 156
- Filter time constant 149
  
- Final Positioning 62
- Following error. 67
- Fuse FS1 position 34
  
- Handshaking 31
- Hardware configured switches 84
- Hardware Scaling 25
- Home 8
- Human interactive mode 37
  
- Immediate 57
- Immediate Commands 39
- In position 64, 122
- In Position Indicator 18
- Indexer counter 71
- Indexer status flags 70
- Individual Commands 38
- Input status 87
- Inputs 10
- Inputs 1 to 7 7
- Integral Action 65
- Integral gain 148
- Interactive operation with a PC/PLC 54
  
- Jog acceleration 88
- Jog velocity 89
  
- Limit status 107
- Loop Commands 43
  
- Manual Tuning Procedure 150
- Motor resolution 66
  
- Non-volatile RAM 50
  
- Optically isolated 10
- Oscillation 149, 154
- Output O1 9
- Output O2 9
- Output O3 9
- Outputs 10
- Overshoot 155
  
- Parameter settings 152
- Pause Commands 45
- Position error 155
- Position error window 150
- Positioner address 33

Power on time 113  
Power-on sequence mode 135, 139  
Preset following index mode 25, 29  
Proportional gain 68, 148

Registration 20

Sequence 49  
Sequence buffer 40  
Sequence buffers 50  
Sequence identifiers 50  
Sequence upload 139  
Servo Control Loop 73  
Servo error flags 114  
Servo loop 148  
Software Scaling 26  
Software version number 115  
Status Request Commands 45  
Stiffness 152  
Stop Move Commands 44

Terminal/computer communication 149  
Tuning 150

Undefined drive fault 33  
Universal (U) 57

Velocity feedback gain 156  
Velocity gain 148, 152

Windows terminal emulator 35  
Write-protect sequences 51

XON, XOFF 31

ZDDC 33