CE

V*i*X250IM & V*i*X500IM
Microstepper
RS232 Indexer Drive
User Guide

Part No: 1600.324.01 July, 2003 (For software revision 2.0 onwards)

IMPORTANT INFORMATION FOR USERS

Installation and Operation of Motion Control Equipment

It is important that motion control equipment is installed and operated in such a way that all applicable safety requirements are met. It is your responsibility as an installer to ensure that you identify the relevant safety standards and comply with them; failure to do so may result in damage to equipment and personal injury. In particular, you should study the contents of this user guide carefully before installing or operating the equipment.

The installation, set-up, test and maintenance procedures given in this User Guide should only be carried out by competent personnel trained in the installation of electronic equipment. Such personnel should be aware of the potential electrical and mechanical hazards associated with mains-powered motion control equipment - please see the safety warning below. The individual or group having overall responsibility for this equipment must ensure that operators are adequately trained.

Under no circumstances will the suppliers of the equipment be liable for any incidental, consequential or special damages of any kind whatsoever, including but not limited to lost profits arising from or in any way connected with the use of the equipment or this user guide.

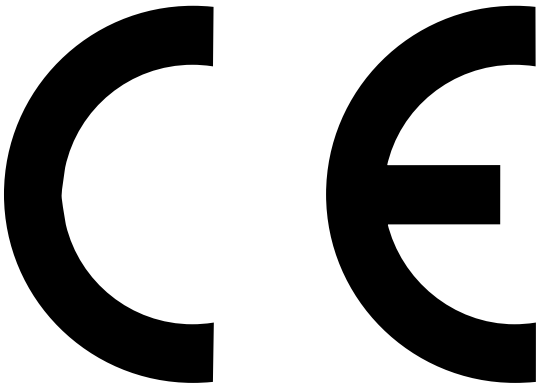
SAFETY WARNING

High-performance motion control equipment is capable of producing rapid movement and very high forces. Unexpected motion may occur especially during the development of controller programs. ***KEEP WELL CLEAR*** of any machinery driven by stepper or servo motors. Never touch any part of the equipment while it is in operation.

This product is sold as a motion control component to be installed in a complete system using good engineering practice. Care must be taken to ensure that the product is installed and used in a safe manner according to local safety laws and regulations. In particular, the product must be enclosed such that no part is accessible while power may be applied.

This and other information from Parker-Hannifin Corporation, its subsidiaries and authorised distributors provides product or system options for further investigation by users having technical expertise. Before you select or use any product or system, it is important that you analyse all aspects of your application and review the information concerning the product in the current product catalogue. The user, through its own analysis and testing, is solely responsible for making the final selection of the system and components and assuring that all performance, safety and warning requirements of the application are met.

If the equipment is used in any manner that does not conform to the instructions given in this user guide, then the protection provided by the equipment may be impaired.



Product Type: **ViX250IM, ViX500IM**

The above product is in compliance with the requirements of directives

- **73/23/EEC** **Low Voltage Directive**
- **93/68/EEC** **CE Marking Directive**
- **89/336/EEC** **Electromagnetic Compatibility Directive**

Provided the installation requirements described in this user guide are met, and there are no special requirements of the installation and operating environment so that the application may be considered typical, the ViX servo drive series installation will conform to the protection requirements of Council Directive 89/336/EEC as amended by Directive 92/31/EEC on the approximation of the laws of the Member States relating to Electromagnetic Compatibility when operated and maintained as intended.

In assessing the overall compliance of an installation consideration must also be given to the effects of mains harmonics and flicker when interfacing the total supply system to the public low voltage supply system.

In accordance with IEC 61800-3:1997 (Adjustable speed electrical power drive systems) this product is of the restricted sales distribution class which meets the needs of an industrial environment when installed as directed. However, further measures may need to be taken for use of the product in a domestic environment.

Compliance is demonstrated by the application of the following standards:

BS EN 61800-3 (1997) including Amendment A11	Adjustable speed electrical power drive systems Part 3. EMC product standard including specific test methods
BS EN 61000-6-2 (2001)	Electromagnetic compatibility – Part 6-2: Generic standards Immunity for industrial environments
BS EN 61000-6-4 (2001)	Electromagnetic compatibility – Part 6-4: Generic standards – Emission standard for industrial environments
BS EN 61010-1 (1993) including Amendment A2	Safety requirements for electrical equipment for measurement, control, and laboratory use. Part 1. General requirements

WARNING – Risk of damage and/or personal injury

The ViX drives described in this user guide contain no user-serviceable parts. Attempting to open the case of any unit, or to replace any internal component, may result in damage to the unit and/or personal injury. This may also void the warranty.

Contact Addresses

For engineering assistance in Europe:
Parker Hannifin plc
Electromechanical
Division - Digiplan
21 Balena Close
Poole, Dorset
England, BH17 7DX
Tel: +44 (0)1202-699000
Fax: +44 (0)1202-695750
e-mail: sales.digiplan@parker.com
e-mail: support.digiplan@parker.com
Website: www.parker-emd.com

For engineering assistance in Germany
Parker Hannifin GmbH
Electromechanical
Division - Hauser
P. O. Box: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg, Germany
Tel: +49 (0)781 509-0
Fax: +49 (0)781 509-176
e-mail: sales.hauser@parker.com
e-mail: techhelp_emd_OG@parker.com
Website: www.parker-emd.com

For engineering assistance in Italy
Parker Hannifin SpA
Divisione SBC
20092 Cinisello Balsamo
Milan,
Italy Via Gounod, 1

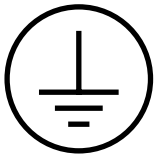
Tel: +39 02 6601 2478
Fax: +39 02 6601 2808

e-mail: sales.sbc@parker.com
Website: www.parker-emd.com

For engineering assistance in the U.S.:
Parker Hannifin Corporation
Compumotor Division
5500 Business Park Drive, Suite D
Rohnert Park
CA 94928
USA
Tel: (800) 358-9070
Fax: (707) 584-3793
FaxBack System: (800) 936-6939
e-mail: CMR_help@parker.com
Website: www.compumotor.com

Symbols used, have the following meanings:

Caution -
Refer to the
accompanying documentation



Protective conductor terminal

Contents

The ViX250IM/500IM Microstepper Indexer Drive is UL-Recognised under file E194158. This means it may be incorporated into end-user products that may be eligible for UL Listing, Classification or Certification.

User Guide Issue Change Summary

This user guide, version 1600.324.01, is the first version of the ViX250IM/ViX500IM Microstepper Indexer Drive.

When a user guide is updated, the new or changed text is differentiated with a change bar in the outside margin (this paragraph is an example). If an entire section is changed, the change bar is located on the outside margin of the section title. For the latest (most up-to-date) changes required by this issue of user guide see the **Latest Changes Sheet** over the page.

Latest Changes Sheet

This page lists important changes occurring immediately before publication or between issue updates:

1. Introduction

Product Description

Available in two current ratings, these microstepper indexer drives employ an optimised digital field oriented current loop to provide low speed smoothness coupled with high speed torque. Advanced digital techniques result in reduced settling time and reduced mid speed instability when compared with similar competitive drive types.

The common use of EASI-V programming language and similar supply requirements make this drive ideal for mixed technology applications when used with the ViX digital servo.

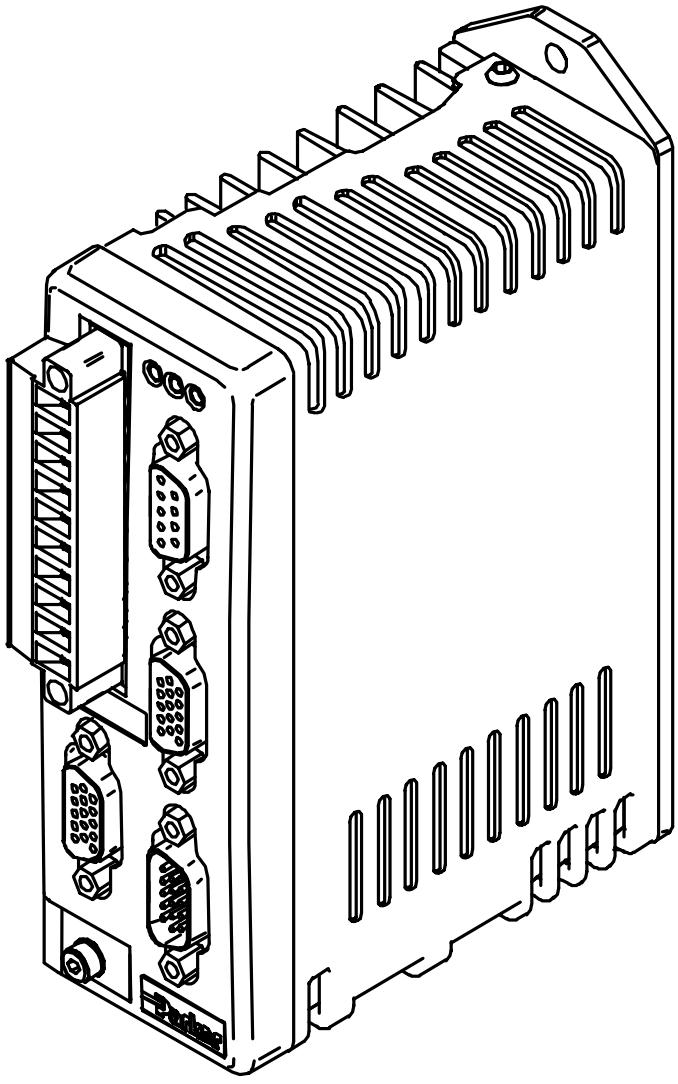


Figure 1-1. ViX250/ViX500 Microstepper Indexer Drive

Product Variants

Digital microstepper indexer drives are available in two current ratings with two interface options. Table 1-1 lists the possible combinations:

Product Code	Description
ViX500IM	5.6A RMS (8A peak) microstepper indexer drive with an RS232 control interface
ViX250IM	2.8A RMS (4A peak) microstepper indexer drive with an RS232 control interface
ViX500CM	5.6A RMS (8A peak) microstepper indexer drive with Canbus/RS485 interface
ViX250CM	2.8A RMS (4A peak) microstepper indexer drive with Canbus/RS485 interface

Table 1-1. ViX250/ViX500 Microstepper Indexer Drive Options

Note: RS485 serial communication is only included in the CANopen version of the drive.

Product Features

Protection Circuits

Motor short circuits, phase to phase, phase to ground
Over-voltage trip
Under-voltage trip
Drive/motor Over-temperature
24V reverse supply protection

Function Indicators

Drive Status/Feedback Fault (HV/FB)
Drive Fault (DF)
Comms. Status (CS)

Outputs and Inputs

3 digital outputs
5 digital inputs
1 analogue input

Fit Kits

A fit kit is available for ViXIM drives:

VIX-KIT		
Part Number	Quantity	Description
1650.937.01	1	Information sheet
5004.023	1	Plastic bag
5006.211	1	Product label
0405.811	1	10-way Flange plug strip
0405.961	1	9-way D-type plug
0405.962	2	15-way HD D-type plug
0405.963	1	15-way HD D-type socket
0409.530	4	9-way D-type cover
0313.020	1	H8FE1115NC ferrite sleeve
4005.218	1	3:1 heatshrink 19mm diam.
4216.101	1	Closed P-clip 9mm ID
4216.102	1	Closed P-clip 10.7mm ID
4216.103	1	Closed P-clip 12.3mm ID

Further Information

This user guide contains all the necessary information for the effective use of this drive. However, to gain a more in-depth understanding of drive applications and motion control, consider attending one of our world-wide Customer Specific Training Workshops, details of which are on our web site (www.parker-emd.com) under Sales & Services, Training.

Examples of previous courses that have proved to be of benefit include:

Use and programming of the DIN rail H & L series drives
PDFX training
Using the 6K controller
EASI Tools programming
Mechanical product training for ET/ER, XR and HPLA

2. Mechanical Installation

Installation Requirements

Environment

ViX drives operate in a temperature range of 0° to 40°C with natural convection, or 50°C Max with forced-air cooling (see Hardware Reference), at normal levels of humidity (5-95% non-condensing). The drives can tolerate atmospheric pollution degree 2, which means only dry, non-conductive pollution is acceptable.

Drive Cooling

Cooling of all drive types is by natural convection up to 40°C. To assist cooling, drives should be installed vertically in an area where there is at least a 50mm (minimum) air gap above and below the package and a 10mm (minimum) gap either side. Avoid mounting heat-producing equipment directly below a drive.

Installers must ensure that the air temperature entering the drive or rising up to the drive is within the ambient temperature restrictions. Under normal use the air temperature leaving the drive and heatsink may be 25°C above ambient.

In the final installation, check that the ambient temperature specification of 40°C Max (without forced air cooling) is not exceeded directly below the top-most drives and that any circulating air flow is not being blocked from reaching the drives. For cabinet cooling calculations, allow 20W per drive. For DIN rail mounting, see the thermal limitations statement in ***Drive Mounting Options***.

Drive Dimensions

ViX250 and ViX500 drives share the same dimensions, shown in Figure 2-1.

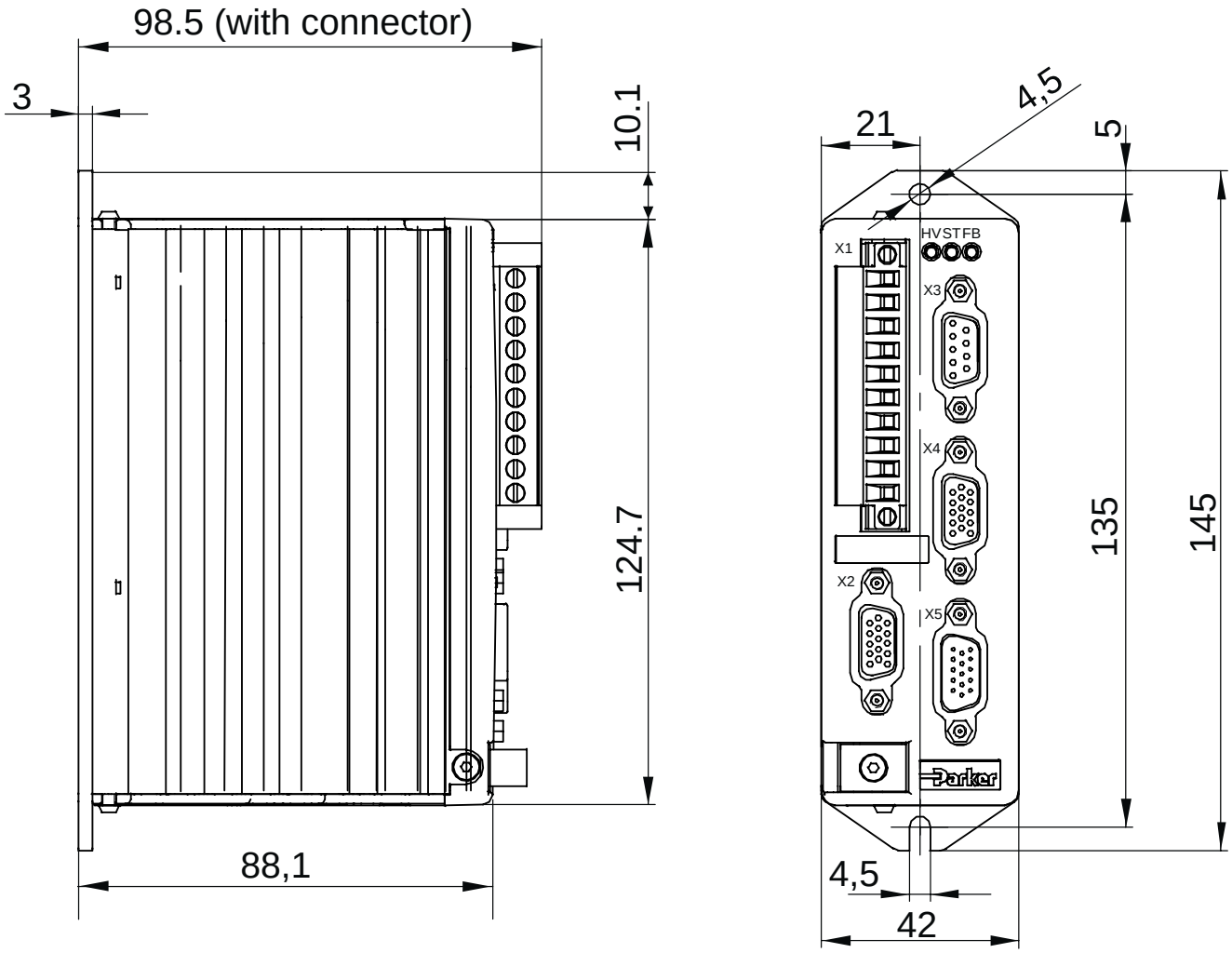


Figure 2-1. ViX250 & ViX500 Dimensions

Drive Mounting Options

If you require a DIN-Rail mounting ViX drive use the optional ***DIN-Rail clip*** adapter bracket shown in Figure 2-2.

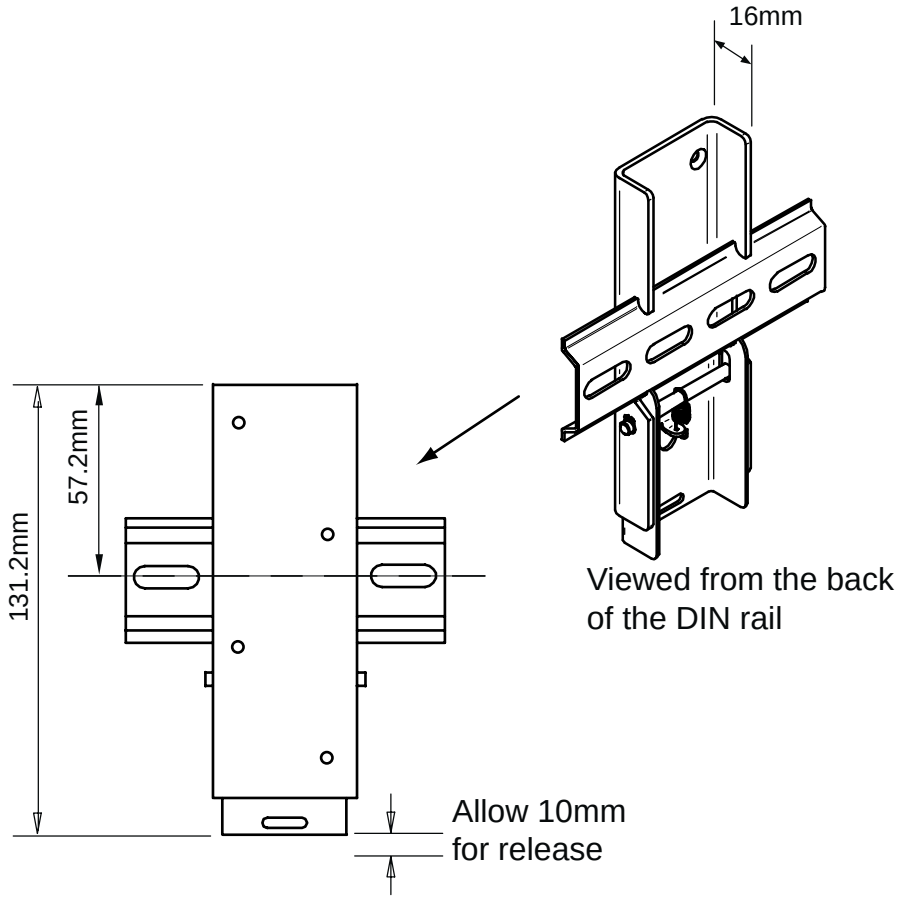


Figure 2-2. DIN-Rail Adapter Bracket

Remove the panel mounting plate from the back of the drive and attach the bracket to the back of the drive using the screws provided. The drive and bracket can now be fixed to a DIN rail by hooking the top of the bracket over the top of the DIN rail and gently pushing the drive forward to engage the lower section of the bracket. Remove the bracket by inserting a flat bladed screwdriver into the release slot to pull down the bottom of the bracket, releasing it from the DIN rail.

Motor Mounting Mechanical Considerations

Keep motors securely fixed in position at all times. Do not test a motor/drive combination without first securing the motor – see the Safety Warning at the front of this user guide.

CAUTION – risk of equipment damage

Do not back drive the motor, that is use the motor in an application that causes mechanical rotation of the motor shaft in a manner uncontrolled by the drive.

Back driving the motor at high speed may damage the drive.

3. Electrical Installation

Installation Safety Requirements

ViX stepper drives meet the requirements of both the European LVD & EMC directives when installed according to the instructions given within this section. It is recommended the drive be installed in an enclosure to protect it from atmospheric contaminants and to prevent operator access while it has power applied. Metal equipment cabinets are ideally suited for housing the equipment since they can provide operator protection, EMC screening, and can be fitted with interlocks arranged to remove all hazardous motor and drive power when the cabinet door is opened. Do not arrange interlocks to open circuit the motor phase connections while the system is still powered, as this could cause damage to the drive.

Precautions

During installation, take the normal precautions against damage caused by electrostatic discharges. Wear earth wrist straps. A switch or circuit breaker must be included in the installation, which must be clearly marked as the disconnecting device and should be within easy reach of the machine operator.

Cabinet Installation

To produce an EMC and LVD compliant installation we recommend that drives are mounted within a steel equipment cabinet. This form of enclosure is not essential to achieving EMC compliance, but does offer the benefits of operator protection and reduces the contamination of the equipment from industrial processes.

A steel equipment cabinet will screen radiated emissions provided all panels are bonded to a central earth point. Separate earth circuits are commonly used within equipment cabinets to minimise the interaction between independent circuits. A circuit switching large currents and sharing a common earth return with another low level signal circuit could conduct electrical noise into the low level circuit, thereby possibly interfering with its operation. For this reason so called ‘dirty earth’ and ‘clean earth’ circuits may be formed within the same cabinet, but all such circuits will eventually need to be returned to the cabinet’s main star earth point.

Mount the individual drives and EMC filter on a metal earth plane. The earth plane will have its own individual star point earth which should be hard wired (using an insulated copper conductor) back to the cabinet’s ‘clean earth’ connection point.

LVD - Low voltage directive
EMC – Electro Magnetic Compatibility directive

Power Supply Connections

Power drives from a DC supply derived from an isolating transformer or a DC power supply (See ***Power Supply Options*** later in this section).

Note: Pin 10 is at the top of the connector X1 and pin 1 at the bottom.

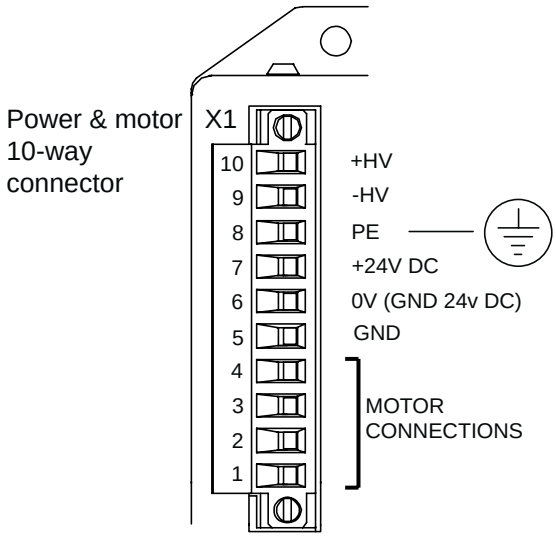


Figure 3-1. X1 Power Connections

WARNING – Possible drive damage

If you use Parker XL Series stepper drives, do not attempt to use any power wiring harness taken from an XL drive. Although the same mating connector is used for both an XL and a ViX, the ViX wiring is the reverse of the XL and the wrong wiring connection will damage the drive.

Mating connector type is: Wieland 8213B/10 F OB, Part number 25.323.4053.0 (Parker part number 0405.811).

Supply Requirements

Power the ViX drives from DC supplies as specified below:

Volts

Drive Type	DC Supply Voltage between +HV and -HV
ViX500	48V to 80V (recommended)
ViX250	24V to 80V

Table 3-1. Drive Supply Voltages

WARNING

The drive HV supply input is not reverse polarity protected.
Reverse polarity connections will damage the drive.

Current and Capacitance

A supply must have a minimum amount of capacitance to support a drive at peak power draw.

Drive Type	DC Supply Current	Supply Capacitance
ViX500	5.6A RMS	6600µF
ViX250	2.8A RMS	3300µF

Table 3-2. Drive Supply Currents

+24V Requirements

Both drive types require a +24V controller and logic supply. The supply may also be required for an encoder and a Fieldbus Expansion Module (FEM).

Absolute voltage range	20 to 27V
Nominal drive current	250mA (excluding encoder, & FEM)
Encoder supply loading	150mA (if required)
FEM current	50mA

Safety Earth Requirements

Earth the drive using the earth pin on X1 (pin 8).

Power Supply Options

A set of torque curves (Figure 3-2) for various motor/drive combinations can be used for calculating an applications likely power requirements.

Higher torque/current requirements will need to use the ViX500 drive and a high current linear supply, such as the PL1100. Further power supply information is given in Appendix A.

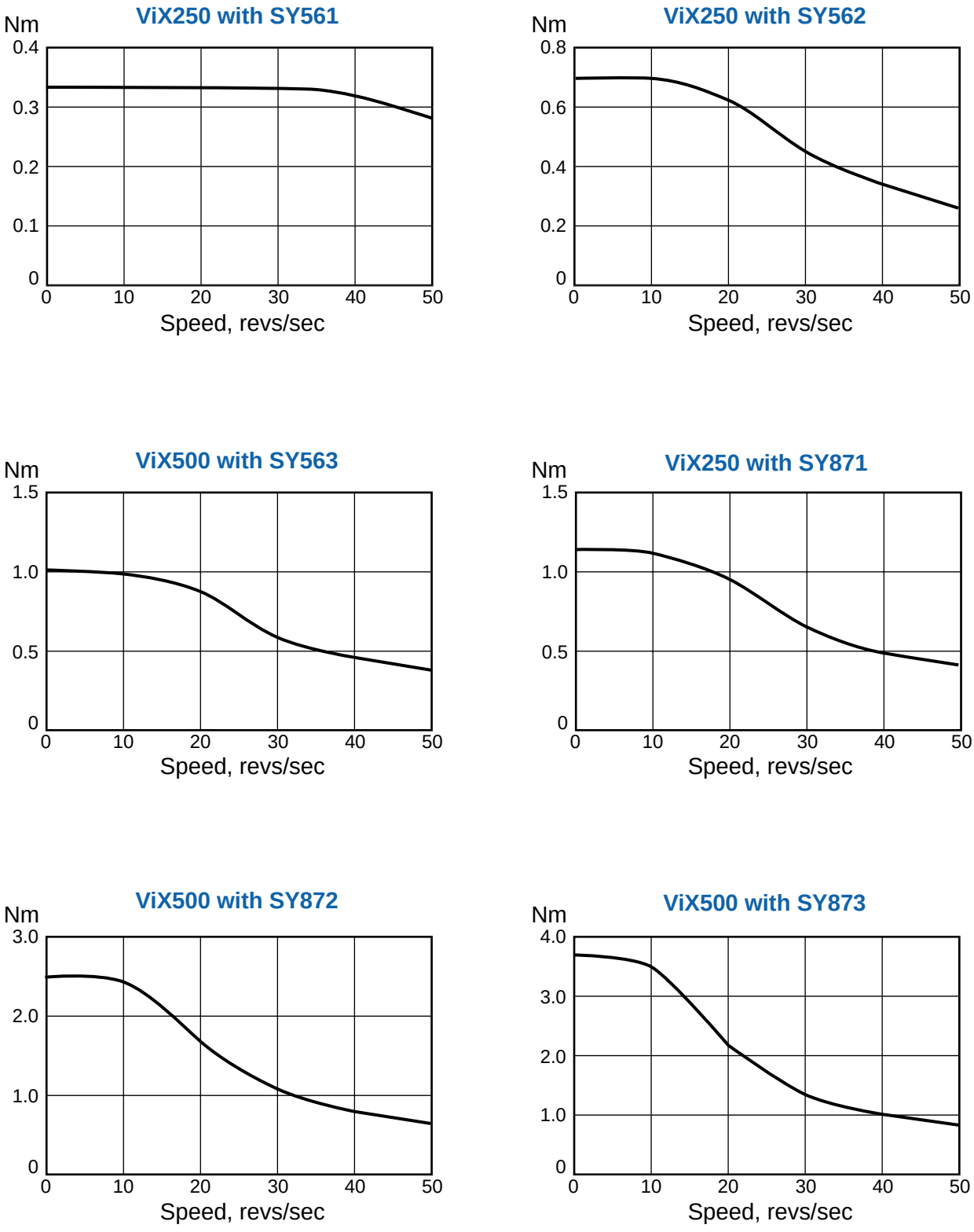


Figure 3-2. Stepper Drive Torque/Speed Data

XL-PSU Power Supply

The XL-PSU is a 250W, power factor corrected, switched mode power supply. Designed for direct operation from world wide single phase AC input voltages, the supply is capable of powering up to two ViX250 drives (see note 1) without the need for an EMC mains input filter (see note 2). The use of the XL-PSU offers the following benefits:

- Auto-adapts to supplies between 95 and 264V AC
- No external EMC filter required
- Compact size
- Built-in +24V DC supply

Note 1: Check the application's power requirements from the torque/speed curve of the motor used.
Note 2: For drives with up to 30 metre motor leads.

For full installation instructions see the XL Power Supply leaflet 1600.300.XX.

XL-PSU Supply/Drive Connections

When used to supply up to two drives the power supply can be wired as shown in Figure 3-3.

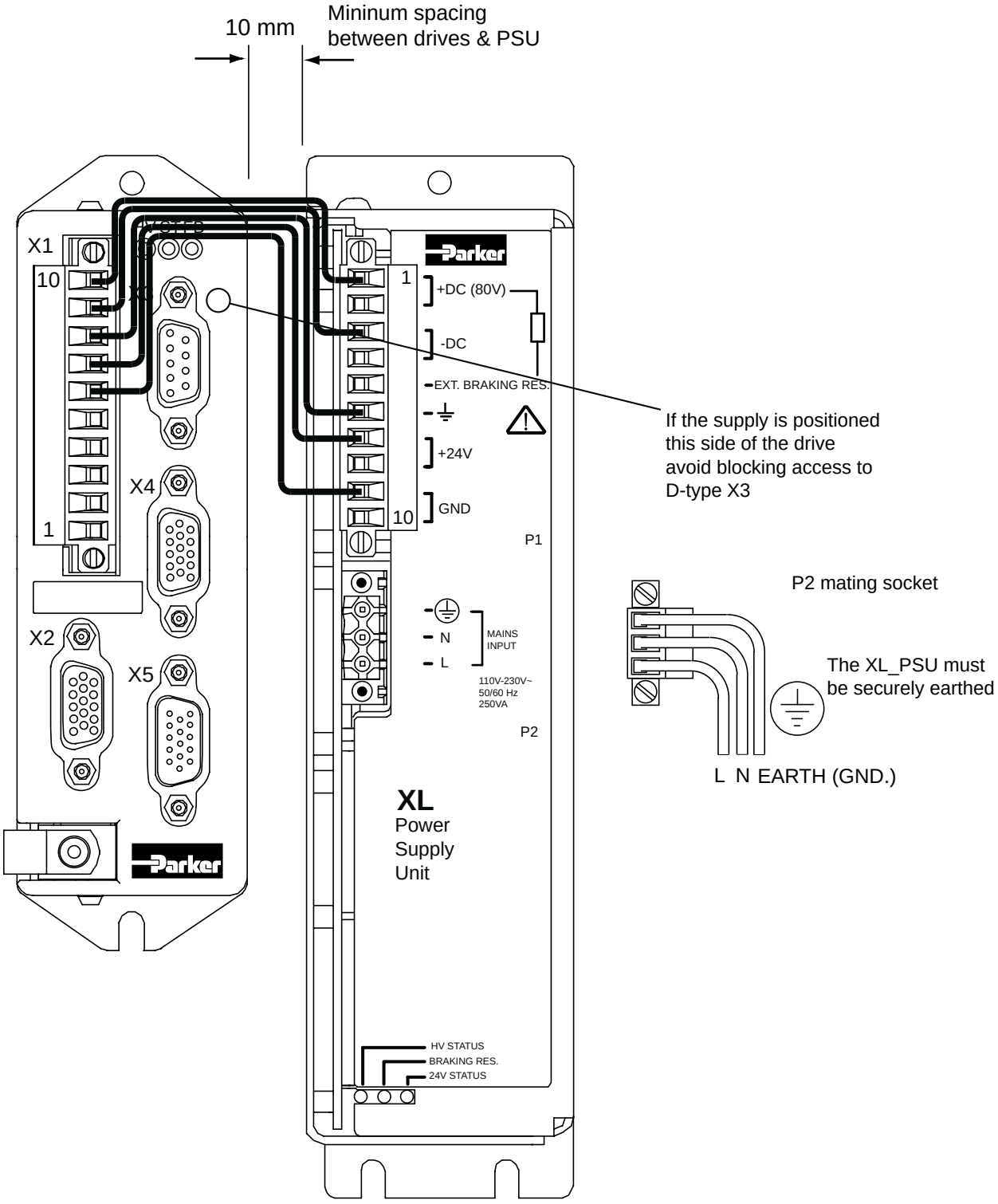


Figure 3-3. XL Power Supply and Drive Connections

Note: A kit of five connecting links is available, called ‘XL-connect’. You will need one kit for every drive.

XL-PSU Mounting Information

Mount the supply vertically, near the drives it will supply. Both the top 4.5mm diameter fixing hole and the bottom two 4.5mm width fixing slots should be used.

Allow a minimum free space of 50mm both below and above its case and 10mm free space on both sides.

Do not mount the supply above or close to other products that generate a significant amount of heat by radiation or convection.

PL1100 Power Supply

General Description

The PL1100 is a linear power supply with a rated output of 1120W (80V/14A) for use with ViX and XL series drives. The supply requires a suitably rated transformer supplying 50V AC RMS for the HV and 20V AC RMS for the +24V DC. The use of the PL1100 offers the following benefits:

- Provides 80V HV and +24V DC output
- Single or three phase operation
- Built-in power dump switch
- Integral fusing

Figure 3-4 shows the PL1100 output wiring for two ViX drives. This illustrates how to route the main HV supply separately to each drive. The lower current requirements of the +24V logic/brake supply can allow the wiring to be linked between drives.

For full installation instructions see the PL1100 Power Supply leaflet 1600.323.XX.

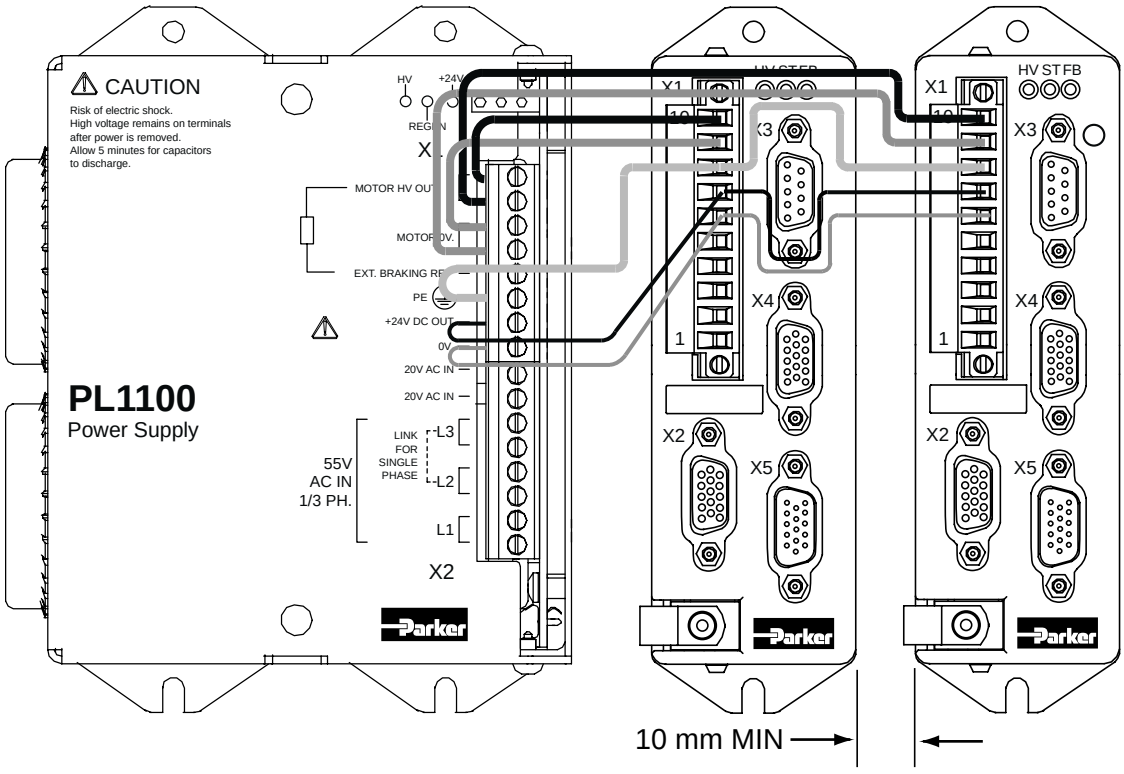


Figure 3-4. PL1100 Power Supply and Drive Connections

EMC Installation

These EMC installation recommendations are based on the expertise acquired during the development of compliant applications, which Parker believes are typical of the way, a drive or drives may be used. Provided you have no special installation requirements or untypical operating environment requirements, ViX drives will conform to current EMC Directives, as defined at the front of this user guide.

General Requirements

ViX mounted drives, unless used with an XL-PSU, will require an EMC supply filter to meet EMC installation compliance requirements. Mount the drive on a conductive panel which is shared with the EMC filters. If the panel has a paint finish, it will be necessary to remove the paint in certain areas to ensure filters and drive make a good large-area metal to metal contact between filter case and panel.

Mount filters close to the drive and keep the supply wiring as short as practical. Attempt to layout the wiring in a way that minimises cross coupling between filtered and non-filtered conductors. This means avoiding running wires from the output of a filter close to those connected to its input. Where you wish to minimise the cross coupling between wires avoid running them side-by-side one another, if they must cross, cross them at 90° to each other. Keep wiring supported and close to cabinet metalwork.

Recommended EMC filter types are CORCOM 6FC10 for loads up to 6A and 3VK1 for the +24V supply up to 3A. Multi-axis systems may require higher current rated filters.

+24V Supply Connections

ViX drives not using an XL-PSU will require a logic supply of +24V DC at 250mA (nominal) per drive. The +24V powers the controller and I/O circuits. Keeping the +24V independent of the drive's internal high voltage bus supply allows the option of keeping the I/O and controller active when no main supply is present.

Connect the +24V supply to X1 pin7 and the return to X1 pin6, the total wire length, from supply to drive, must not exceed 10m.

Connect the +24V supply 0V line to system earth (0V) at some convenient point before the EMC filter input, as shown in the recommended EMC layout diagram, Figure 3-5.

The 24V supply to each drive should be fitted with a time-delay fuse, rated at 3A. Note: The +24V supply used must meet the voltage requirement specification of +24V DC +10% -15%, ripple <1V p-p.

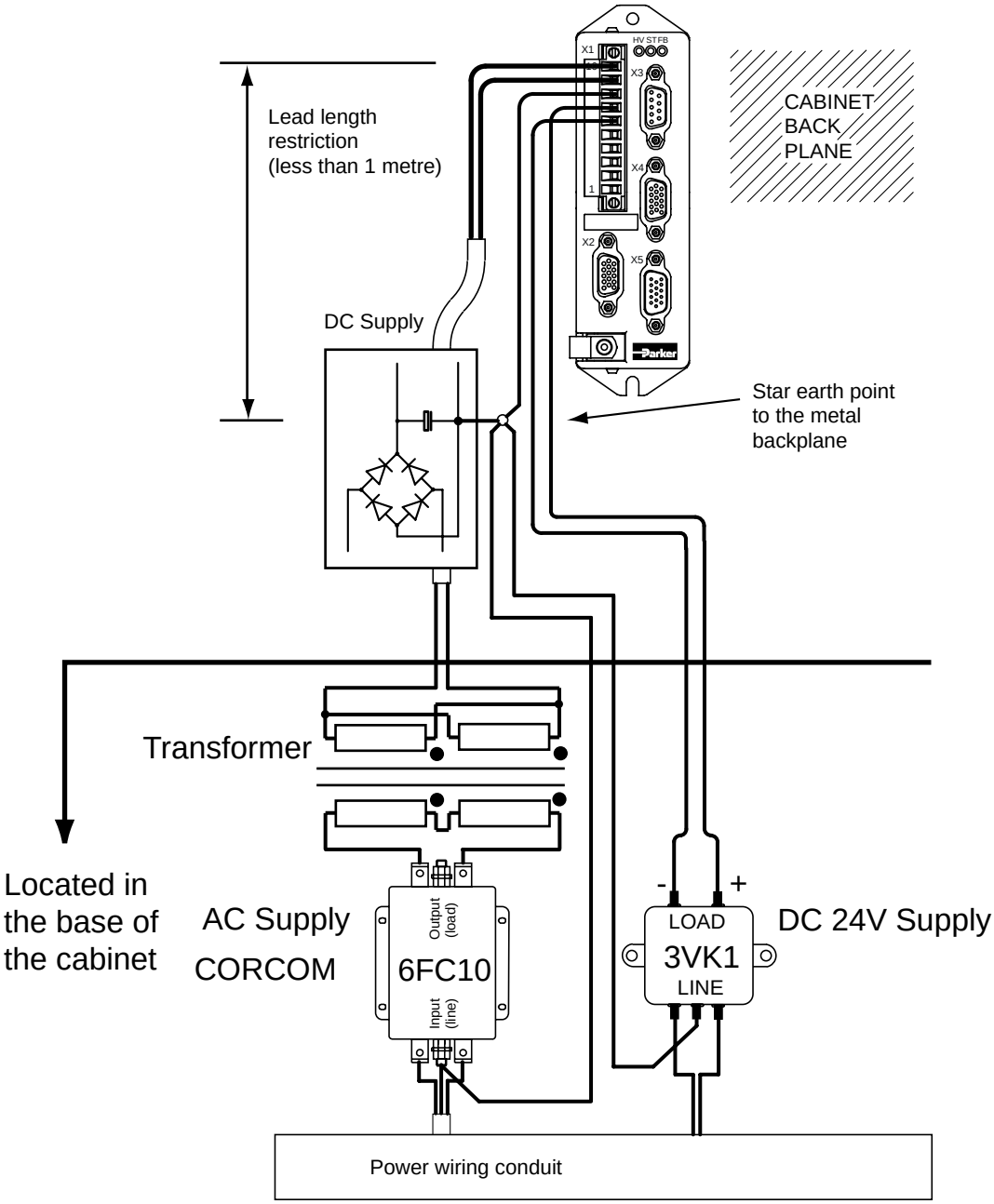


Figure 3-5. ViX EMC Installation

Motor Connections to the Drive

The recommended wire size for ViX250IM/500IM motor cables, of length less than 20m, is 1mm^2. For motor cable lengths greater than 20m (up to a maximum of 50m) use a wire size of 2.5mm^2. Use a cable containing five conductors plus the braided screen (such as Lapp 34805), the green wire being used to provide an earth return to the drive. Termination at the motor must be made using a 360° bond to the motor body, and this may be achieved by using a suitable clamp. Many stepper motors are designed to accommodate an appropriate terminal gland which can be used for this purpose.

At the drive end of the cable, a 360° connection to the screen should be made using the P-clip provided beneath the motor connector. The P-clip needs to be firmly clamped to the copper braid. If the connection appears loose, fold the braid back on itself to increase the amount of braid under the clip and re-tighten.

Custom cables will require the cable insulation to be removed to expose the braided screen. If you are using a motor cable with 2.5mm^2 conductors the size of the P-clip will need to be 9mm to accommodate the increased cable diameter. A ferrite absorber, with a specification matching that of the Chomerics H8FE-1115-NC, is also required to be positioned on the motor cable using heat shrink sleeving or cable ties. The position of the absorber should be within 150mm of the drive. Always secure the cable using the P-clip, as shown. Do not rely upon the connector alone holding the motor cable in place. **Avoid stress on the X1 connector by hanging cables, as this may lead to connector over-heating.**

Make a 360° connection to the screen using one of the stainless steel or brass P-clips supplied within the fit kit.

Size	Parker part number
9mm ID	4216.101
10.7mm ID	4216.102
12.3mm ID	4216.103

Table 3-3. P Clip sizes

Three different size ‘P’ clips allow the use of a variety of motor power cables from different manufactures.

There must be no break in the 360° coverage that the screen provides around the cable conductors. If a connector must be used it should retain the 360° coverage, possibly by the use of an additional metallic casing where it passes through the bulkhead of the enclosure. The cable screen must not be bonded to the cabinet at the point of entry. Its function is to return high-frequency chopping current back to the drive. This may require mounting the connector on a sub-panel insulated from the main cabinet, or using a connector having an internal screen which is insulated from the connector housing. Within the cabinet itself, all the motor cables should lie in the same trunking as far as possible. They must be kept separate from any low-level control signal cables. This applies particularly where the control cables are unscreened and run close to the drive.

Note that the motor cable routing within the equipment cabinet should be kept at least 300mm away from I/O cables carrying control signals.

All motor connections must be made using a high quality braided-screen cable. Cables using a metallised plastic bandage for an earth screen are unsuitable and in fact provide very little screening. Care must be taken when terminating the cable screen, the screen itself is comparatively fragile; bending it round a tight radius can seriously affect the screening performance. The selected cable must have a temperature rating which is adequate for the expected operating temperature of the motor case.

Motor Cables

Motor cables may be ordered using the part numbers listed in Table 3-4.

Product code/Part number	**Length (metres)**
STC20-0300	3
STC20-0500	5
STC20-1500	15

Table 3-4. Motor Cables

Motor Phase Contactors

We recommend that motor phase contactors are not used within the motor power cables. As an alternative, make use of the drive's power stage 'enable' control signal.

Ferrite absorber specifications

The absorbers described in these installation instructions use a low-grade ferrite material that has high losses at radio frequencies. They therefore act like a high impedance in this waveband. Produced by Parker Chomerics, the recommended component is suitable for use with cable having an outside diameter up to 10mm. The specification is as follows:
Chomerics part number H8FE-1115-NC (Parker part number 0313.020)
Outside diameter 17.5mm
Inside diameter 10.7mm
Length 28.5mm
Impedance at 25MHz 80 ohm
Impedance at 100MHz 120ohm
Curie temperature 130°C (the device should not be operated near this temperature)

Motor Selection

Usually optimum performance will be obtained when the current rating of the motor is between 1 and 1.5 times the drive rating. Drives can be de-rated to accommodate motors with lower current ratings (using variable MC within the MOTOR command), however the high speed torque will be reduced.

Do not use a drive setting which gives an output current greater than the motor rating.

With 4 lead motors the bipolar rating is quoted and this should match the criteria stated above.

With 8 lead motors the bipolar rating of the motor, which is normally quoted, refers to a parallel winding connection. With the windings connected in series the current rating of the motor connection will be 50% that of the bipolar rating, and the motor will give improved low-speed torque, but reduced high-speed torque.

The ViX250IM/ViX500IM will drive motors having an inductance as low as 0.5mH and as high as 20mH, but the recommended motor inductance range is between 0.8mH and 10mH.

Performance of the ViX250/ViX500IM is optimised for the following motor types, listed in Table 3-5.

Motor Type	Motor Rated Current in Amps*	Motor Inductance in mH per phase*	ViX500IM	ViX250IM
SY561	4.2	1.0		✔
SY562	4.2	2.6	✔	✔
SY563	6.5	1.2	✔	
SY871	4.2	1.6	✔	✔
SY872	6.5	1.5	✔	
SY873	8.4	1.7	✔	
SY1072	8.0	2.4	✔	

*(parallel connection)

Table 3-5. SY Optimum Motor Types

Motor Voltage Ratings

Motors with a withstand voltage rating from phase to earth of 1000V AC should be used. An insulation withstand rating of 500V AC is acceptable if an isolating transformer with earthed screen is used to power the system, and 0V input is earthed, as specified.

Large Motors

The largest recommended motor size is a 34-frame 3-stack. Please contact Parker EMD if you wish to use a larger frame motor.

Motor Connections at the Motor

Motor connections should be made directly between the drive and motor, the use of any switching devices, such as contactors is not recommended.

In the majority of applications the drive will be used with an eight lead motor with the windings connected in parallel or series, as shown in Figure 3-6. Motor connections will need to be determined from the motors data sheet. These are normally identified by wire colour or terminal markings, depending upon the make of the motor.

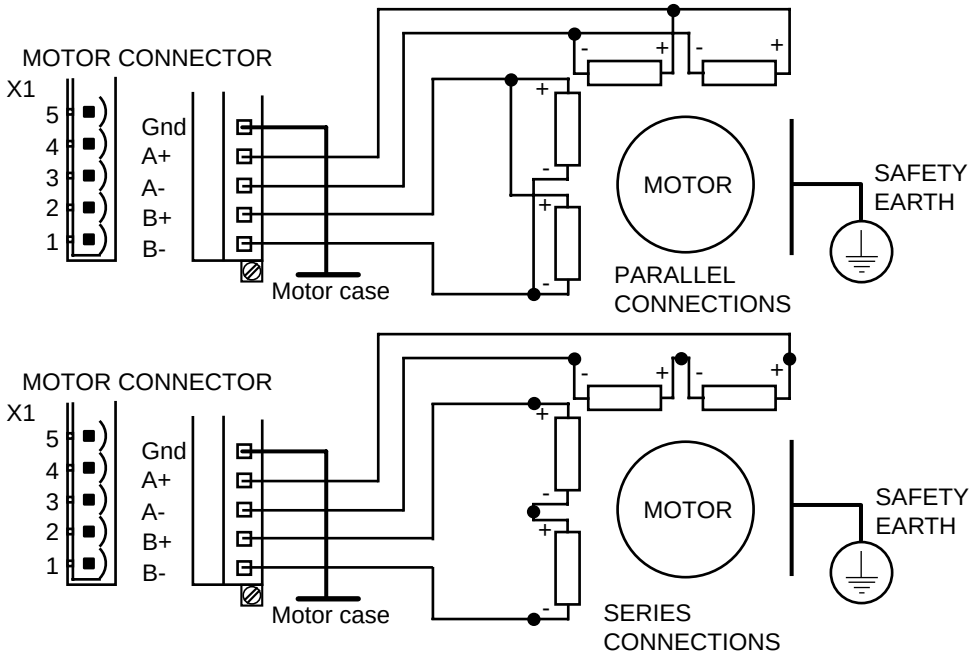


Figure 3-6. 8 Lead Motor Connection Options

WARNING - High Temperature

The motor case temperature may exceed 70°C and should be guarded from operator contact.

Motor Safety Earth/Ground Connection

It is recommended that the motor is independently bonded to a local safety earth point. The safety earth lead should be at least 2.5mm^2 in area.

Custom Motor Set Up

Within screen 2 of **Guided stepper initialisation**, clicking upon the Setup custom button will open the window shown in Figure 3-7.

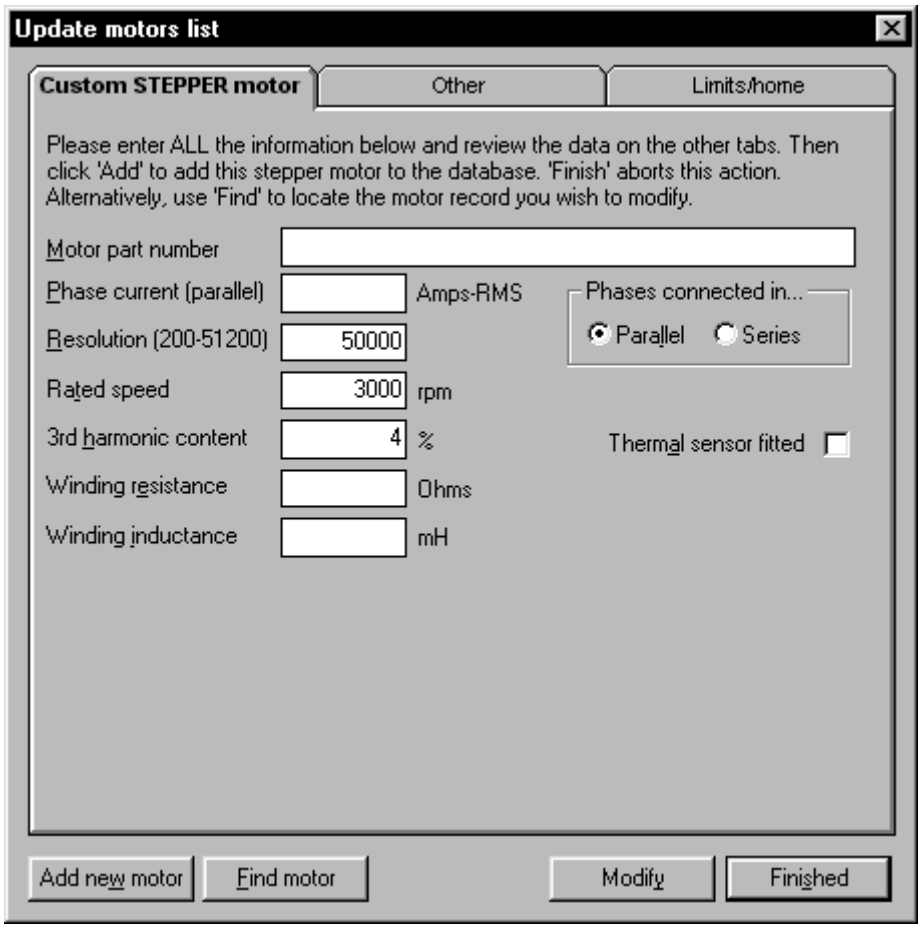


Figure 3-7. EASI-V Custom Motor Configuration Window

Motor	the general name/number for the motor.
Phase current (parallel)	continuous current rating of the motor in Amps RMS.
Resolution	number of steps per revolution
Rated speed	shaft speed in rpm for a rotary stepper.
Winding resistance	resistance of a single phase winding measured line-to-line in Ohms.
Winding inductance	inductance of a single phase winding measured line-to-line in mH.

The Other Parameters Tab

Selecting the Other parameters tab gives you access to the screen shown in Figure 3-8.

Figure 3-8. EASI-V Custom Motor Other Parameters

In-position time (IT)	
Digital I/O	The decimal number required by the IC system variable to configure the input/output state of the drive.

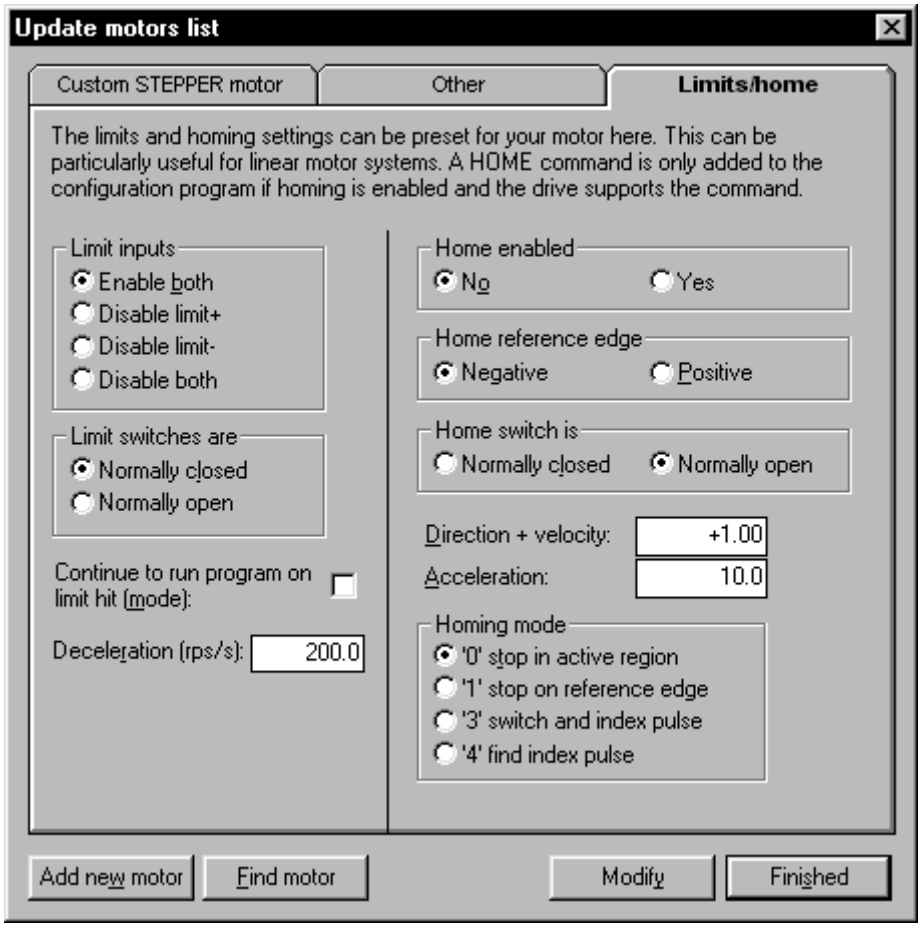


Figure 3-9. EASI-V Custom Motor Limits/home Parameters

Limit inputs	Four radio buttons used to configure the limit inputs.
Limit switches	Selection of normally closed or normally open limit switches.
Home enabled	Enable/disable the HOME command.
Home reference edge	Select the required edge of the home switch where you wish the home position to be.
Home switch	Defines the type of home switch used, normally open or closed.
Direction + velocity	Required direction and velocity. Positive direction commands must produce movement towards the positive limit.
Acceleration	Acceleration of the motor in revs/s/s.
Homing mode	Homing mode selection – see sub-section on homing for an explanation of these modes.

Motor Voltage Ratings

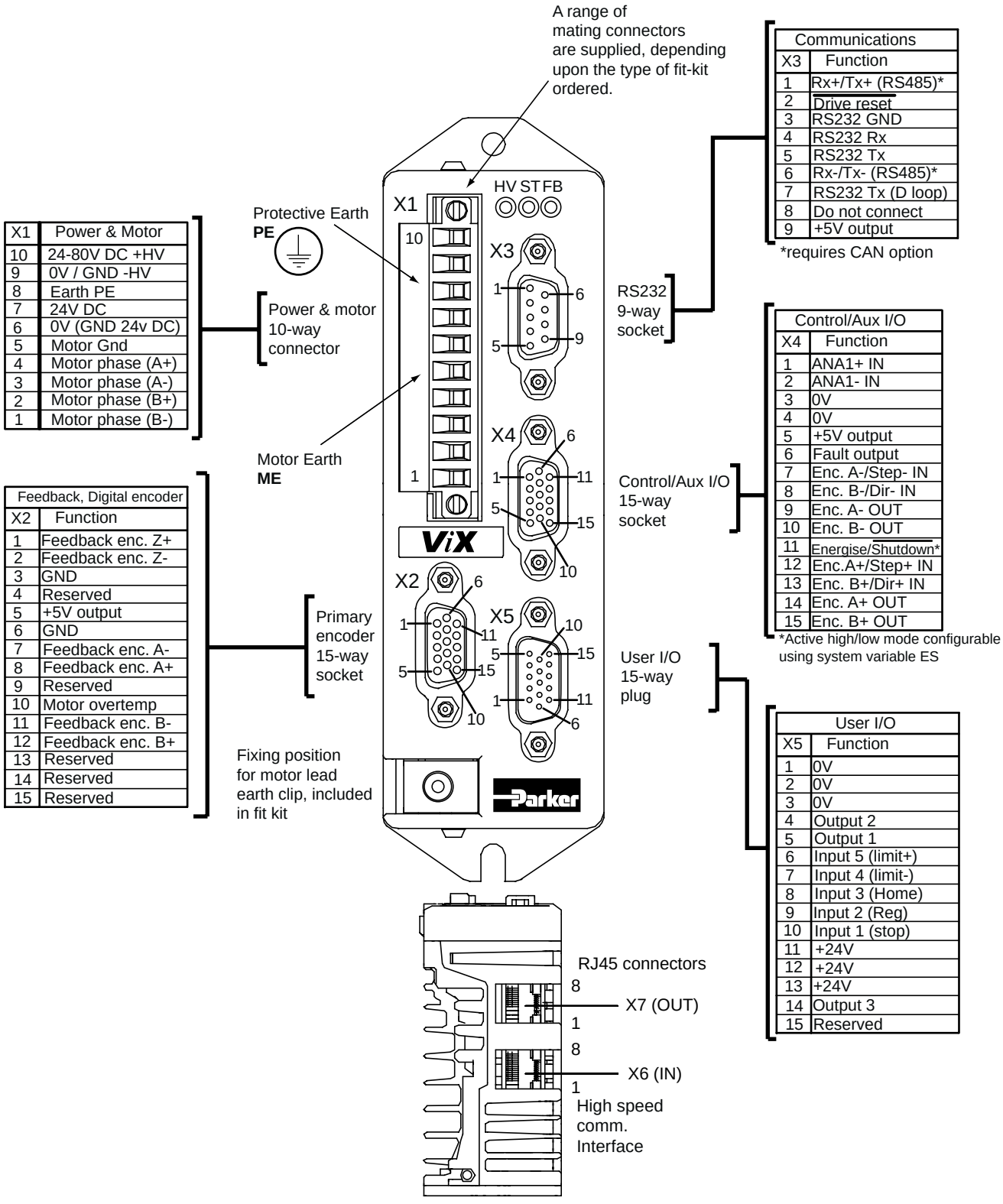
Motors with a withstand voltage rating from phase to earth of 1000V AC should be used. An insulation withstand rating of 500V AC is acceptable if an isolating transformer with earthed screen is used to power the system, and X1 pin9 (0V/GND) input is earthed, as specified.

Motor Safety Earth/Ground Connection

It is recommended that the motor is independently bonded to a local safety earth point. The safety earth lead should be at least 2.5mm^2 in area.

Short Circuit Protection

The motor outputs are protected against overload and short circuits.



X1	Power & Motor
10	24-80V DC +HV
9	0V / GND -HV
8	Earth PE
7	24V DC
6	0V (GND 24v DC)
5	Motor Gnd
4	Motor phase (A+)
3	Motor phase (A-)
2	Motor phase (B+)
1	Motor phase (B-)

Feedback, Digital encoder	
X2	Function
1	Feedback enc. Z+
2	Feedback enc. Z-
3	GND
4	Reserved
5	+5V output
6	GND
7	Feedback enc. A-
8	Feedback enc. A+
9	Reserved
10	Motor overtemp
11	Feedback enc. B-
12	Feedback enc. B+
13	Reserved
14	Reserved
15	Reserved

Communications	
X3	Function
1	Rx+/Tx+ (RS485)*
2	Drive reset
3	RS232 GND
4	RS232 Rx
5	RS232 Tx
6	Rx-/Tx- (RS485)*
7	RS232 Tx (D loop)
8	Do not connect
9	+5V output

*requires CAN option

Control/Aux I/O	
X4	Function
1	ANA1+ IN
2	ANA1- IN
3	0V
4	0V
5	+5V output
6	Fault output
7	Enc. A-/Step- IN
8	Enc. B-/Dir- IN
9	Enc. A- OUT
10	Enc. B- OUT
11	Energise/Shutdown*
12	Enc.A+/Step+ IN
13	Enc. B+/Dir+ IN
14	Enc. A+ OUT
15	Enc. B+ OUT

*Active high/low mode configurable using system variable ES

User I/O	
X5	Function
1	0V
2	0V
3	0V
4	Output 2
5	Output 1
6	Input 5 (limit+)
7	Input 4 (limit-)
8	Input 3 (Home)
9	Input 2 (Reg)
10	Input 1 (stop)
11	+24V
12	+24V
13	+24V
14	Output 3
15	Reserved

Figure 3-10. ViX Connector Pin Layout

Terminal Description

X1 Connector

X1 is the main power and motor connector. Both HV, +24V and the motor phase connections are made to X1.

Connector Type

The mating connector for X1 is a Wieland 8213B/10F, part number 25.323.4053.0 (Parker part number 0405.811). An approval marked version of this connector has the part number 25.323.1053.0.

Connector Pin Out

Connector Pin X1	Signal Name
10	24 to 80V DC +HV
9	0V/GND -HV
8	Earth PE
7	24V DC
6	0V (GND for 24V DC)
5	Motor Earth
4	Motor phase (A+)
3	Motor phase (A-)
2	Motor phase (B+)
1	Motor phase (B-)

Table 3-6. X1 Power and Motor Connections

Motor Connections at the Drive

Refer to the EMC installation information earlier in this section.

X2 Connector

X2 provides the primary input connections for the motor feedback device. This is the input that should be used for position maintenance and stall detection functions.

Connector Type

Connector type is a high-density 15-way D-type socket.

Connector Pin Out

Connector Pin X2	Primary Encoder
1	Feedback enc. Z+
2	Feedback enc. Z-
3	GND
4	reserved
5	+5V output
6	GND
7	Feedback enc. A-
8	Feedback enc. A+
9	reserved
10	Motor overtemp+.
11	Feedback enc. B-
12	Feedback enc. B+
13	reserved
14	reserved
15	reserved

Table 3-7. X2 Primary Feedback Connections

Encoder Compatibility for X2 & X4

Incremental channels Input specification

Signal format quadrature 5V differential signals (A+, A-, B+, B-) index mark (Z+, Z-).
Maximum digital encoder input frequency 2.0MHz pre quad, 8.0 MHz post quadrature.
Maximum encoder supply current 350mA.

Motor Overtemperature Sensor

Standard Parker stepper motors do not use an over-temperature sensor, however when using custom motors provision is made for the connection of either a thermal switch or thermistor device. The following devices are supported:

- Thermik SNM130ES
- Cantherm F11 110-2-5 U106

Other ptc thermistors with a switch like characteristic are supported to DIN44081/44082.

The input requires a normally closed switch to be connected to GND on X2 pin 3 or 6.

If you use a custom motor with no overtemperature sensor fitted, make sure you leave the 'Thermal sensor fitted' check box un-checked in the ***Custom Motor Set Up*** screen, within Easi-V to prevent an overtemperature fault being reported. This is the default setting in Easi-V.

X3 Connector

X3 is the RS232/RS485 communications connector. RJ45 connectors X6 and X7 may also be used for inter-drive communications where multi-axis systems are used.

RS485 Operation

RS485 operation is only possible on drives fitted with the appropriate FEM (Fieldbus Expansion Module). If you require this feature please order the **ViX – CM** drive type.

Connector Type

Connector type is a 9-way D-type socket.

Connector Pin Out

Connector Pin X3	Function
1	Rx+/Tx+ (RS485)
2	$\overline{\text{drive reset}}$
3	RS232 GND
4	RS232 Rx
5	RS232 Tx
6	Rx-/Tx- (RS485)
7	RS232 Tx (D loop)
8	Do not connect
9	+5V output

Table 3-8. X3 RS232/RS485 Connections

Baud Rate

Use system variable BR to alter the baud rate of serial communications. Any change made to the baud rate will only take effect following a save (SV) and system reset or power cycle.

Reset to RS232 Mode

To reset the drive to RS232 mode and to return to factory settings, remove power from the drive, connect X3 pin 2 to GND and restore power.

CAUTION

This will erase ALL of your user settings and programs in volatile memory. The non-volatile memory will not be overwritten until a save command is issued.

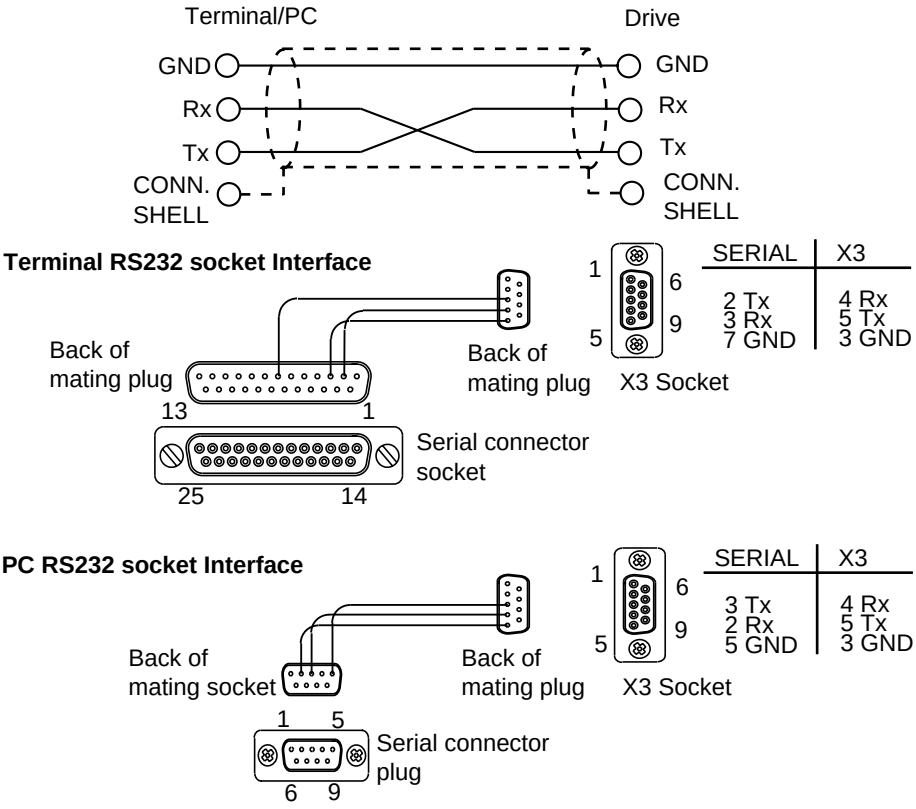


Figure 3-11. X3 D-type Connector RS232 Connections

Inter-drive RS232 Connections

Use the RJ45 connectors X6 and X7 to inter-connect drives, see ***RS232 Daisy Chain*** later in this section. Always make the primary connection via D-type X3.

RS232 Connecting Leads

RS232 cables can be ordered from Parker EMD. Various lengths are available as listed in Table 3-9.

Part Number	Length
RS232-EASI-0250	2. 5m
RS232-EASI-0500	5.0m
RS232-EASI-0750	7.5m
RS232-EASI-1000	10.0m
RS232-EASI-1250	12.5m
RS232-EASI-1500	15.0m

Table 3-9. RS232 Connection Lead Types

X4 Connector

Connector X4 gives access to the **following encoder** input and output signals and the differential analogue inputs. Certain input and output connections are dependent upon the state of system variables EO (Encoder Output) and EI (Encoder Input). Encoder output signals are not generated internally by the drive, they mirror the state of the feedback encoder inputs (if present). Use encoder connection X2 for position maintenance and stall detection feedback.

Connector Type

Connector type is a high-density 15-way D-type socket.

Connector Pin Out

Connector Pin X4	Encoder I/O
1	ANA1+ (input)
2	ANA1- (input)
3	0V
4	0V
5	+5V output
6	Fault
11	Energise/Shutdown_bar* (input)

*See system variable ES

Table 3-10. X4 Encoder I/O Connections

Inputs Depending Upon the State of System Variable EI

Connector Pin X4	EI=0	EI=1	EI=2
12	STEP+	CW+	A+
7	STEP-	CW-	A-
13	DIR+	CCW+	B+
8	DIR-	CCW-	B-

Outputs Depending Upon the State of System Variable EO*

Connector Pin X4	EO=0	EO=1	EO=2
14	STEP+	CW+	A+
9	STEP-	CW-	A-
15	DIR+	CCW+	B+
10	DIR-	CCW-	B-

*Requires encoder feedback input on X2

Differential Analogue Input

The ViX stepper drive can accept a differential analogue input for use with the FRATE command. The input circuit, shown in Figure 3-12, can interface to an external +/-10V differential signal. Analogue to digital conversion (12-bit resolution) converts the analogue input to a digital value for use within the drive. Read the value of the analogue input as a count via system variable AI.

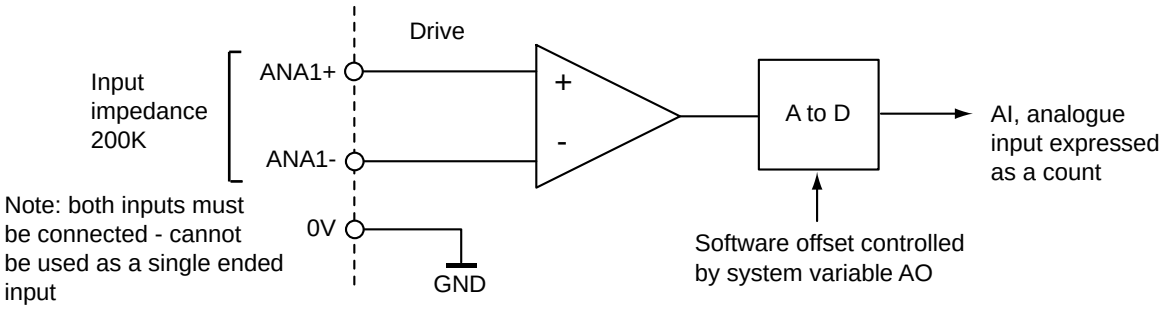


Figure 3-12. Analogue Differential Input

Figure 3-13 shows the input characteristic.

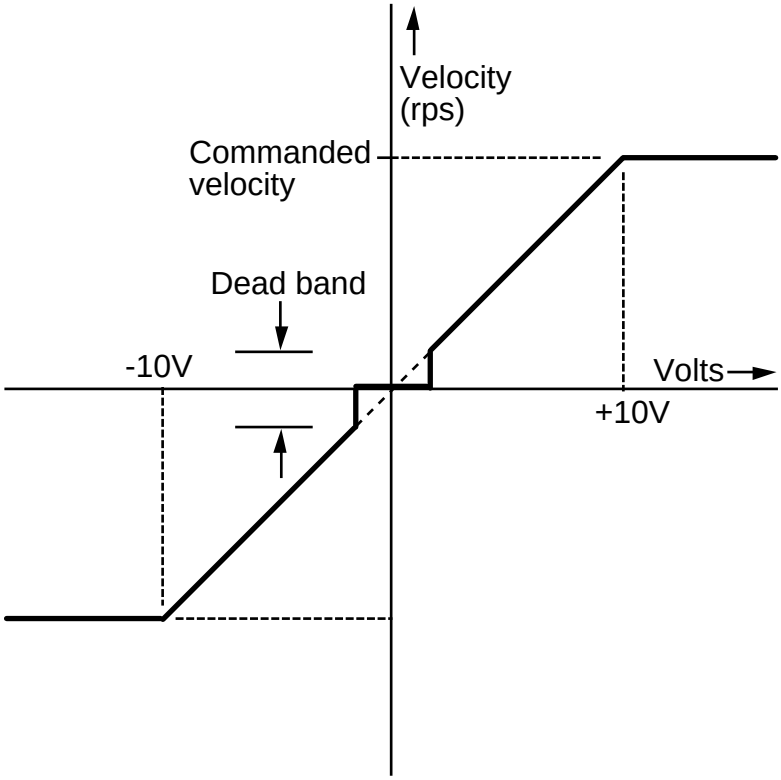


Figure 3-13. Analogue Differential Input Characteristic

An analogue deadband can be set, using system variable ‘AB’.

Energise/Shutdown

Enable the drive by allowing the input pin to float high ‘1’ or by linking the pin to zero volts, depending upon the input’s polarity. System variable ES controls the polarity of this input. The default state of ES (Energise Sense) requires X4 input pin 11 to be connected to 0V to enable the drive.

The function of this input differs when in mode ‘MP’, please refer to the ***Command Reference*** section for more details.

X5 Connector

X5 is the user Input/Output connector.

Connector Type

Connector type is a high-density 15-way D-type plug.

Connector Pin Out

Connector Pin X5	Input/Output
1	0V
2	0V
3	0V
4	Output 2
5	Output 1
6	Input 5 (limit+)
7	Input 4 (limit-)
8	Input 3 (home)
9	Input 2 (registration)
10	Input 1 (stop)
11	+24V
12	+24V
13	+24V
14	Output 3
15	Reserved

Table 3-11. X5 User Input/Output Connections

User Inputs

Inputs can be configured using the Easi-V graphic interface or by writing directly to the IC system variable. By adjusting the user input configuration, you can set the input switching level threshold and you can set the internal input resistor to be a pull-up or a pull-down. Figure 3-14 shows the position of software switches.

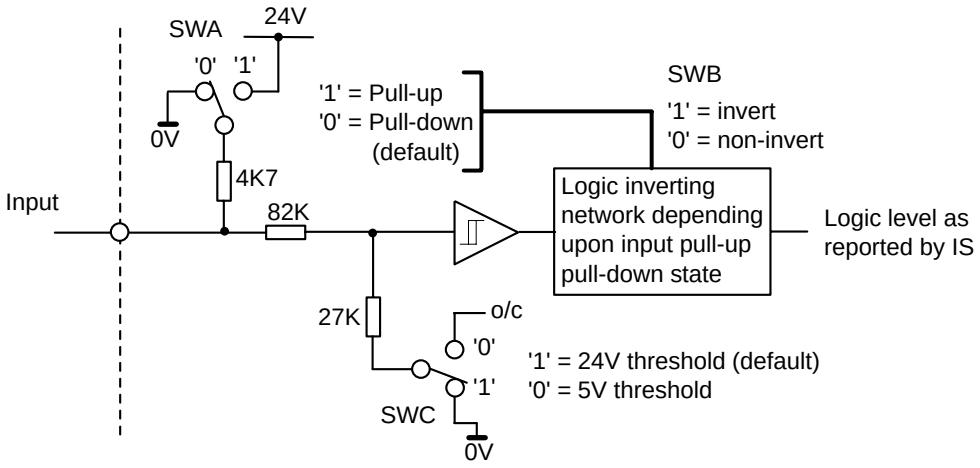


Figure 3-14. User Input Circuit

User inputs are high logic level and low level logic compatible, but must be configured as pull-down inputs when used with low-level 5V logic, since the pull-up mode always pulls-up to +24V.

Only one input is shown above, individual inputs can be set-up on a one-to-one basis allowing different inputs to have different threshold switching levels or different pull-up, pull-down arrangements.

CAUTION – Unexpected motor movement

De-energise the drive before making any changes to the I/O configuration.

User Outputs

User outputs can be configured using the Easi-V graphic interface or by writing directly to the IC system variable. By adjusting the user output configuration, you can set the output to source or sink current. Figure 3-15 shows the output circuit.

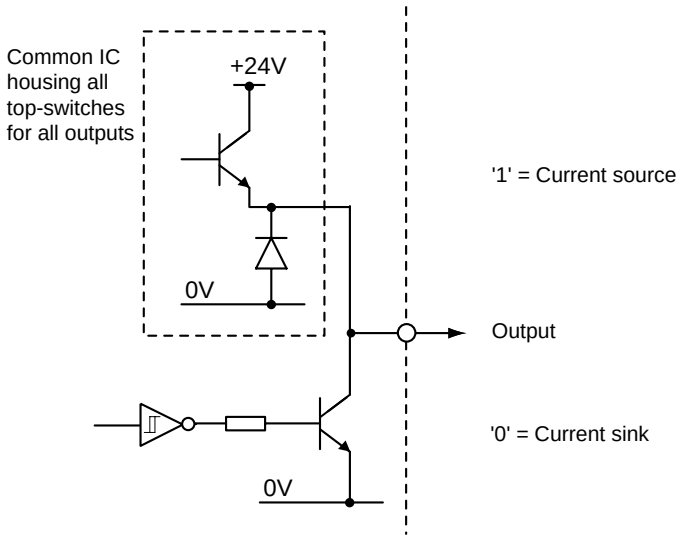


Figure 3-15. User Output Circuit

User outputs are compatible with high-level 24V logic only. Each output can source or sink 50mA.

Note: The easiest way of configuring the drive's inputs and outputs is to use the Easi-V graphic user interface.

Input/Output Configuration

To set-up the input and output configuration without using the EASI-V graphic interface, you will need to write configuration patterns to the two-byte IC parameter, as shown.

aW(IC,{4 digit decimal number equivalent to a two-byte number})

Bits 8 to 12 control the switching threshold of inputs 1 to 5 (SWC setting).
Setting a bit to a '1' gives a 24V switching threshold, a '0' gives a 5V switching threshold.

Bit	15	14	13	12	11	10	9	8
IC content	not used	not used	not used	in_5	in_4	in_3	in_2	in_1

Bits 0 to 4 control the input resistor pull-down/pull-up of inputs 1 to 5 (SWA setting).
Setting a bit to a ‘1’ sets the input resistor to be a pull-up to +24V, a ‘0’ sets the resistor to be a pull-down.
Bits 5 to 7 controls the source/sink operation of outputs 1 to 3.
Setting a bit to a ‘1’ sources current from the +24V rail via the upper half of the output, while setting a bit to a ‘0’ sinks current from a connected input through the lower output transistor to 0V.

Bit	7	6	5	4	3	2	1	0
IC content	out_3	out_2	out_1	in_5	in_4	in_3	in_2	in_1

Note:
[1] SWB is automatically set to ensure that the software will report ‘0’ for a closed input switch and ‘1’ for an open input switch.
[2] sourcing outputs can only be used with 24V high level logic.
[3] 5V tolerant input connections must only be used with pull-down (sink) configuration as the input pull-up always pulls up to 24V.
[4] Invalid combinations will report an error (*E), and the User Fault (UF) bit 1 is set (value out of range).

User inputs are high logic level and low level logic compatible, but must be configured as pull-down inputs when used with low-level 5V logic, since the pull-up always pulls-up to +24V.

Example

Configure a drive with inputs in_1 and in_2 arranged as pull-down 5V threshold logic. In_3, In_4 and In_5 as pull-up high threshold level logic, and all outputs as current sources. The binary pattern required is:

(MSB) (LSB)
00011100 11111100

In hex. this becomes 1CFC, which in decimal is 7420

So the required command to (say) axis 3 is **3W(IC,7420)**

IC default setting

The default setting for the drive is all inputs set to 24V threshold, all inputs pulled-down and all outputs sourcing, which gives a binary pattern of 00011111 11100000, which in hex. gives 1FE0, resulting in the decimal equivalent of 8160.

Fault Output

The fault output is an independent NPN open-collector output which is normally ‘low’, active ‘high’. The output ratings are +30V maximum in the OFF condition and 15mA maximum in the ON condition. Figure 3-16 shows the output circuit.

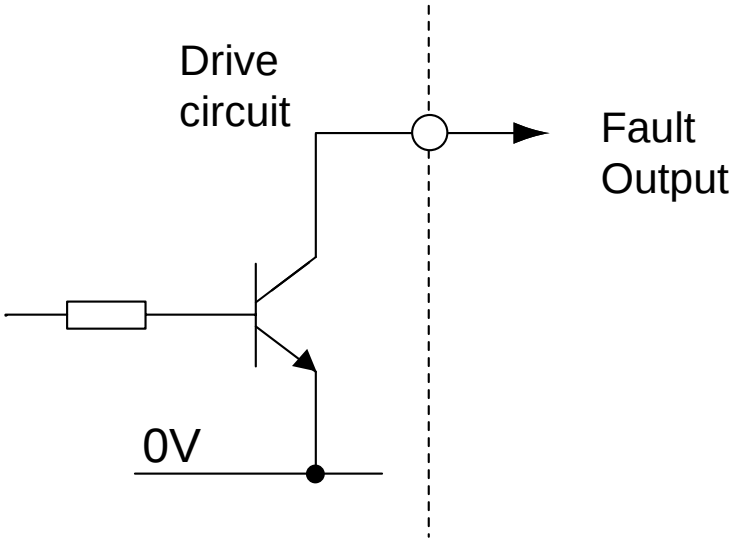


Figure 3-16. Fault Output Circuit

Limit Switches

The drive has two limit inputs, the positive limit input and the negative limit input. When wiring the limit switches it is essential to check that a positive direction command produces motion towards the positive limit switch.

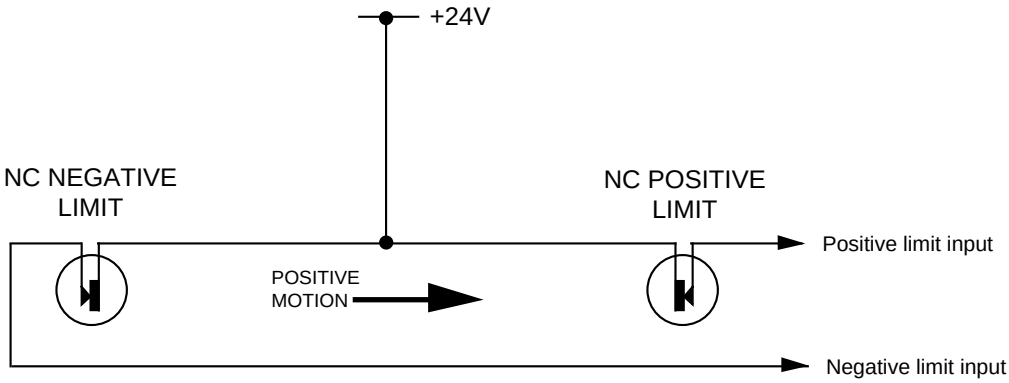


Figure 3-17. Limit and Stop Switch Configuration

RJ45 Interfaces

Positioned beneath the drive are two RJ45 communication interfaces X6 and X7. The two interfaces provide support for Canbus, RS485 (using the Field Expansion Module) and daisy chain ports for multi-axis RS232 connections between drives.

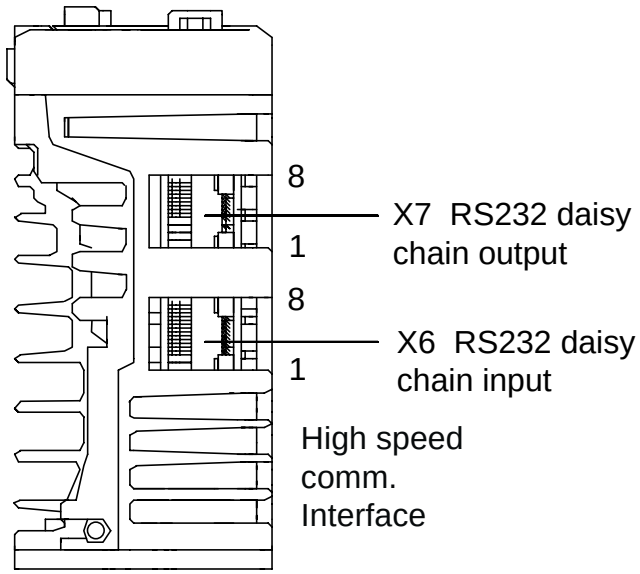


Figure 3-18. Position of Connectors X6 and X7

	FEM1	CAT5 cable colours
X6	**CANopen/RS485**	
1	RX+/TX+ RS485	White/Orange
2	RX-/TX- RS485	Orange
3	CAN H	White/Green
4	RS232 Gnd	Blue
5	RS232 Gnd	White/Blue
6	CAN L	Green
7	RS232 Tx	White/Brown
8	RS232 Rx	Brown

X7		
1	RX+/TX+ RS485	White/Orange
2	RX-/TX- RS485	Orange
3	CAN H	White/Green
4	RS232 sense	Blue
5	RS232 Gnd	White/Blue
6	CAN L	Green
7	RS232 Rx	White/Brown
8	RS232 Tx	Brown

Table 3-12. X6/X7 Input/Output Connections

CAN Bus Termination

Systems using CANopen will need to terminate the final X7 output with a 120 ohms quarter watt resistor connected between X7 pins 3 and 6. A ready-made CAN bus RJ45 terminator is available as shown in Figure 3-19 (Parker part number ‘ViX-RJ45-G).

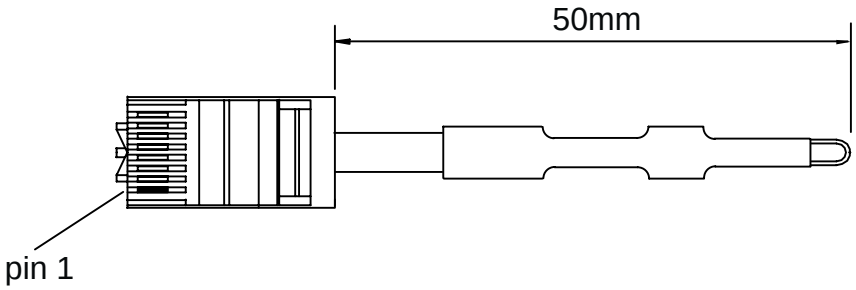


Figure 3-19. CAN Bus Terminator

Communication Daisy Chain

Drives can be 'daisy-chained' for RS232/RS485* operation as shown below. Using this arrangement the drive connected to the controlling PC, via its front panel D-type connector, becomes axis #1. To automatically assign addresses, connect all power, motor, feedback and communication cables then power-up all the drives, see **'#' command** for more details. At the controlling PC, type the following commands:

#1 ;cause the 1st drive to establish the daisy chain
in a 3-axis system the response will be #4
0SV ;save the address configuration
0Z ;reset

response should be a single check sum from axis 1
more than one check sum indicates a problem, possibly no save command

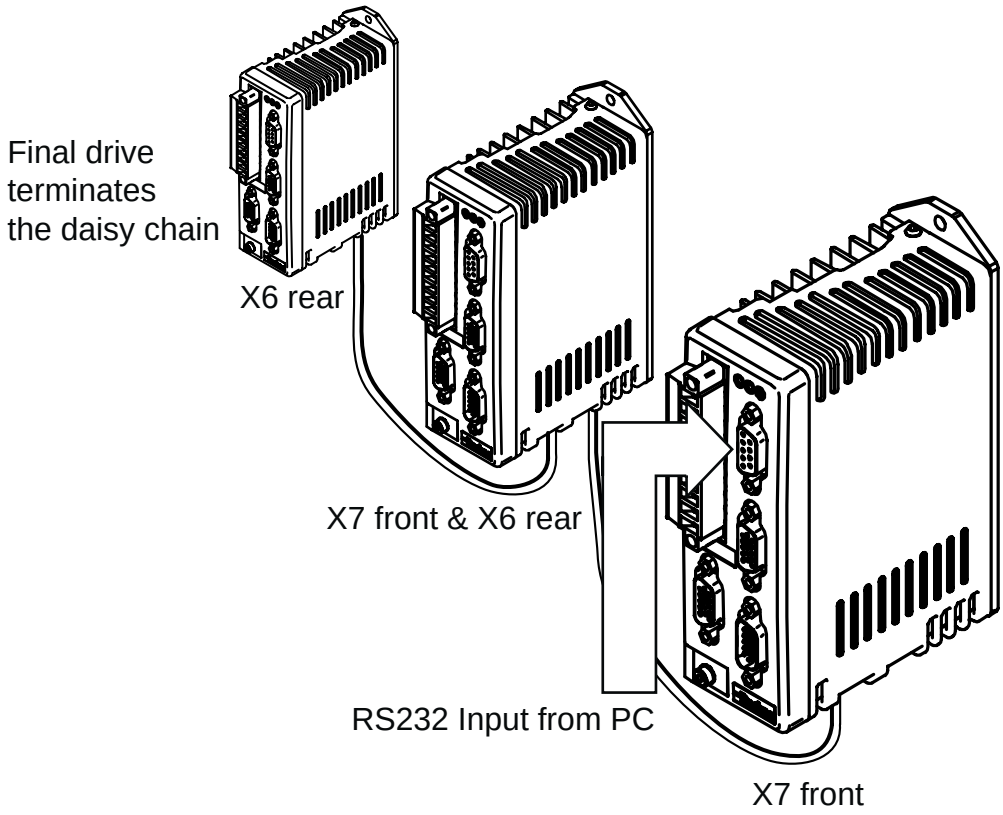


Figure 3-20. RJ45 RS232 Daisy Chain Connections

*Note for RS485 operation, the drive will need to be fitted with a FEM CAN & RS485 interface. Using the command **#1(485)** will switch all drives to 485 operation, which is automatically saved.

Using the X6/X7 connections on the underside of the drive will allow the last drive in the chain to detect that there are no more connections made to X7 which will close the daisy chain loop back internally.

To maintain the integrity of the EMC screening, all RS232 and RS485 connections must be made via the drive's X3 D-type connector.

RJ45 Connecting Leads

RJ45 link cables can be ordered from Parker EMD. Various lengths are available as listed in Table 3-13.

Part Number	Length
VIX-RJ45-0025	0.25m
VIX-RJ45-0050	0.5m
VIX-RJ45-0075	0.75m
VIX-RJ45-0100	1.0m
VIX-RJ45-0200	2.0m

Table 3-13. RJ45 Connection Lead Types

Note: Individual cables that are within the RJ45 daisy chain system must not exceed a length of 2m. Where a cable length greater than 2m is required between axes, a fully screened connection should be made via connector X3.

4. Control of ViX Drives

Overview

This section introduces you to the operation of the ViX stepper drive, the implementation of motion control moves and the way commands are used. Basic controller operation is described together with the code structure. How system information is signalled via system variables and the use of various flag registers for status and fault reporting are described. Both basic and advanced motion control functions are covered including elements of event driven code used for fault reporting and registration.

Controller Operation

ViX intelligent drives have an integrated controller which can be driven directly by a PC over a serial link, or programmed to respond to code selected by event triggers or user instructions.

Direct Mode

Direct operation of the controller over a serial link can be used for program development/downloading purposes or direct on-line control from an industrial PC or PLC. When used directly the controller will accept commands prefixed with the drive's address and will action the commands as they are received. In direct mode any controlling application program is stored in a remote location and is only downloaded to the drive when required.

Programmed Mode

This mode allows a program stored within the drive to control operations. The program can be written off-line on a PC and then downloaded to the drive via a serial link. The application program is stored within the drive and is automatically invoked at power up provided it is enabled by the **<a>ARM1X** command and the program has a START label. Alternatively, you could directly issue a **<a>GOTO(START)** command.

Code Structure

You write program code as a series of blocks. Each code block has a unique label at the beginning and is terminated with an END label (block delimiter). The use of labels allows the code structure of the form illustrated in Figure 4-1, which shows the block nature together with an example of code.

Declare

Declare every label used in a program, apart from START, REG, NOREG and FAULT that have been pre-declared. If a label is declared, but not defined, a runtime error will be signalled when it is called.
Note: START, REG, NOREG and FAULT are all reserved labels.

You can only declare labels in the command line at the start of a program or within the START code. The choice is between memory efficiency and the retention of declared labels

during up-loading/down-loading of programs. Declaring labels in the command line, before any START code, makes the most efficient use of the available memory. If you then up-load the program to a PC and later down-load the same program the declarations will have been lost. To retain declared labels you must declare them in the START code, this allows a program to be up-loaded and down-loaded without loss of declared labels, although more memory will be used. Despite the greater amount of memory being used, it is safer to make the declarations within the START label as there is less chance of forgetting to declare parts of the code.

Example of DECLARE being used in the command line:

```
1K                      ;Kill or stop any program currently running
1CLEAR(ALL)             ;Erase all existing programs
1DECLARE(MAIN)          ;Declare labels
1DECLARE(MOVE1)
1DECLARE(MOVE2)
      .
      .
```

Example of DECLARE being used following the START label:

```
1K                      ;Kill or stop any program currently running
1CLEAR(ALL)             ;Erase all programs
1START:
     1DECLARE(MAIN)     ;Declare labels
     1DECLARE(MOVE1)
     1DECLARE(MOVE2)
          .
1END
```

Labels

Labels consist of up to 5 upper case alphanumeric characters terminated with a colon (:), but a label must begin with an alpha character. Choose a name that is relevant to the operation being performed, or a system label name.
To terminate a code block use ‘END’ (no colon).
You can use up to 20 labels, although four of these have already been allocated to START, REG, NOREG and FAULT, leaving sixteen for general use.

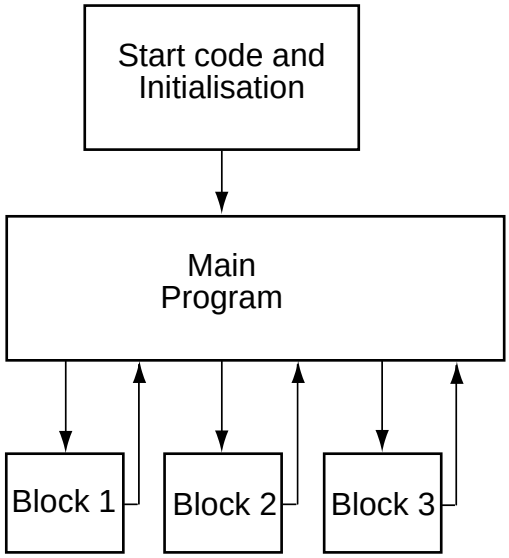
Label Execution

By using the label select command (LSEL), labelled code blocks can be triggered by a digital pattern appearing on certain user inputs. The command defines the user inputs to be used, the style of code detected (BCD or binary) and the manner in which the code is executed (continuous or re-trigger).

Enable the LSEL command using its on/off parameter to allow input selection of labels.

Structure

The code example of an absolute positioning move shown in Figure 4-1 demonstrates how to write code that follows the block structure. Use the start code to initialise the drive:




```
1START:
1DECLARE(MAIN)
1DECLARE(MOVE2)
1LIMITS(3,0,0)
1GOTO(MAIN)
1END

1MAIN:
1PROFILE2(40,10,-48000,25)
1GOSUB(MOVE2)
1END

1MOVE2:
1W(PA,0)
1MA
1USE(2)
1G
1END
```

Figure 4-1. Program Structure

Example:

```
1START:                          ; start label definition
1DECLARE(MAIN)                   ; declare labels
1DECLARE(MOVE2)                  ; declare move 2
1LIMITS(3,0,0)                   ; configure limits.
1GOTO(MAIN)                      ; goto main program
1END
```

Use the MAIN part of the program to define profiles and to control the order of moves:

```
1MAIN:                           ; main label definition
1PROFILE2(40,10,-48000,25)       ; define move parameters
1GOSUB(MOVE2)                    ; jump to label move 2
1END                             ; end of label definition
```

Finally, call individual moves from the main part of the program:

```
1MOVE2:                 ; define program label “move2”
1W(PA,0)                ; zero position absolute
1MA                     ; absolute positioning move
1USE(2)                 ; use motion profile 2
1G                      ; execute move
1END                    ; end of program move 2 definition
```

Note: PROFILE2 defined in the main part of the program has the following characteristics:

ACCELERATION 40rps^2 , DECELERATION 10rps^2, DISTANCE 48000 steps (12 REVS MOVE), NEGATIVE DIRECTION , VELOCITY 25 rps.

In small programs, the start code can be combined with the main part of the program. For experienced X-code users, the shorter blocks of code in the example above, accessed via subroutines, is the equivalent of a sequence.

A second example illustrates the code required for an incremental move. Here the START and MAIN code blocks have been combined within the START block:

```
1START:                 ; start label definition
1DECLARE(MOVE1)         ; declare move1 label
1LIMITS(3,0,0)          ; configure limits (disable, n/c).
1PROFILE1(80,20,24000,20)  ; define move parameters
1GOTO(MOVE1)            ; transfer to label move 1
1END                    ; end of label definition

1MOVE1:                 ; define program label.
1MI                     ; incremental positioning move
1USE(1)                 ; use motion profile 1
1G                      ; execute move
1END                    ; end of program move 1 definition.
```

Note: [1] **DEVICE ADDRESSING IS REQUIRED FOR ALL COMMANDS**

[2] PROFILE1 has the following characteristics:

ACCELERATION 80rps^2 , DECELERATION 20rps^2, DISTANCE 24000 steps (6 REVS MOVE), POSITIVE DIRECTION , VELOCITY 20 rps.

LOOP Command

The block structure of the code lends itself to performing repetitive operations, using the LOOP command. The command can be used to call a particular labelled block of code for either a specified number of times or continuously.

An example using the LOOP command is given below, again the START and MAIN code blocks have been combined within the START block:

```
1START:                     ; start label definition
1DECLARE(LOAD)              ; declare label
1LIMITS(3,0,0)              ; disable limits
1PROFILE3(100,50,4000,35)   ; define move parameters
1MI                         ; set mode to incremental
1LOOP(LOAD,6)               ; repeat the load unload 6 times
1END                        ; end of label definition

1LOAD:                      ; define program label load
1USE(3)                     ; use motion profile 3
1O(XX0)                     ; ensure o/p 3 is off
1T1                         ; wait for 1 sec delay
1G                          ; execute move
1O(XX1)                     ; turn on o/p 3
1T1                         ; wait for 1 sec delay
1END                        ; end of label definition
```

Reserved System Labels

Certain pre-defined labels are recognised by the controller as containing code used for common operations. If event triggered code is enabled (ARM1), the code entered for these common operations will be automatically run when the event occurs.
System labels have the following names:

START: specifies the power on code, run using the ARM1 command
FAULT: specifies the code that is to be run when a fault occurs
REG: specifies the code to be run when a registration mark is detected within the registration window
NOREG: specifies the code to be run when a registration mark is not detected within the registration window

Note: If necessary, these labels can be used for other purposes, but cannot be re-named.

Fault Label

Use the pre-declared label named FAULT to identify a block of code that is executed when a particular problem (fault) has been detected. The code following the FAULT label needs to change the state of an output, to indicate a fault has occurred and then go on to possibly diagnose the problem. Once the problem has been corrected, the FAULT code will need to detect an external 'reset', by monitoring a designated input and then execute an ON command to clear the FAULT. At the end of the FAULT code a GOTO(START) can be issued to restart the program. **This style of programming will always ensure that once a fault is detected the drive will stop and will not start again until commanded to do so.**

Before the code following a FAULT label can be executed certain conditions must be met, these are:

- FAULT must be defined
- ARM must be set to enable a FAULT label

This means FAULT label code must be present and the **ARMX1** command exists at the beginning of the code.

The conditions under which the FAULT label is called will vary depending upon the fault itself and the condition of various other commands and command parameters. An exact description is presented in Table 4-1. However, in general, a FAULT label will be called given any one of the following conditions:

- An attempt to go home further onto a limit is made and the limit is enabled.
- An attempt to go further onto a limit is made with no fault label currently running, the limit configuration is stop on limit and the limit is enabled.
- A limit is hit during motion and the move is not a go home, a fault label is not being run, the limit configuration is stop on limit and the limit is enabled.
- A drive fault has occurred, but no drive programming is taking place.
- When it is called from a GOTO, GOSUB or LOOP command*.

*Note: in this case a FAULT has not actually occurred, consequently the FAULT label will be called irrespective of the state of the ARM command.

Table 4-1 summarises the conditions necessary for the FAULT label to be called. The FAULT label will **not** be called when any one of the following conditions occur:

- There is an error whilst sending a command
- There is a general run time error with the program
- The program memory area becomes full
- A label is attempted to be run when it does not exist
- The transmit buffer or receive buffer suffer an overflow

Fault Condition	**Command & parameter conditions**						
	FAULT label defined	Not GH	Fault ARM bit	Limit is enabled	Not running fault label	Limit decision is stop program execution	Not program-ming the drive
G onto a limit	Y	N/A	Y	Y	Y	Y	N/A
Hit limit	Y	Y	Y	Y	Y	Y	N/A
Drive fault	Y	N/A	Y	N/A	N/A	N/A	Y
GOTO	Y	N/A	N/A	N/A	N/A	N/A	Y
GOSUB	Y	N/A	N/A	N/A	N/A	N/A	Y
LOOP	Y	N/A	N/A	N/A	N/A	N/A	Y

Table 4-1. Conditions Required to Call a Fault Label

Example

The following example shows the use of a FAULT label within a program.

```
1ARM11                  ;enable auto-run on power-up & enable fault routine
1SV                     ;save the settings

1START:                 ;start of program
1ARM11                  ;re-enable auto-run & fault in case 'K' command sent
.
<initialisation commands>
.
1O(1XX)                 ;turn on output 1 - drive OK
.
<main process commands>
.
1END

1FAULT:                 ;fault routine
1O(0XX)                 ;turn off output 1 - drive fault
.
<diagnostic code - if required>*
.
1TR(IN,=,1XXXX)         ;wait for input 1 to become active (RESET)
1ON                     ;clear fault
1GOTO(START)            ;run from start of program again
1END
```

*Note: An example of diagnostic code is given in the sub-section entitled ***Conditional Code*** later within this section.

Start Label

The system label START: introduces the drive's setup and initialisation code. With ARM enabled the code is automatically executed at system start-up*. Consequently the code needs to be saved with ARM1X set. If you save a program with ARM0X set, the start-up code will not run and the controller will only respond to serial input commands.
*Unless a drive fault is pending and a fault routine is defined and armed.

Start Label Example:

```
1START:
        1"RUNNING"
        -

        -

1END

1FAULT:
        1"FAULT"
        1TR(IN,=,1XXXX)
        1GOTO(START)
1END

1ARM01      ;enable fault routine only
1SV         ;save all settings
```

If you cycle the power to the drive the "**START**" routine will not automatically run. To start it you would have to type in **1GOTO(START)**. However, the "**FAULT**" routine will run if a fault occurs

Entering the following code:

```
1ARM11      ;enable auto run on "START"
1SV         ;save all settings
```

The "**START**" routine should automatically run on the next power-up.

Use of the LSEL Command

You can let user inputs call programmed routines by the use of special label names and associated user input numbers. By including the code you wish to action, following a pre-defined input label, will enable your code to be run when the defined user input is activated. For example, to select one of three labels using two user inputs, the code would be:

```
1START:
1CLEAR(ALL)          ;clear memory
1DECLARE(L1)         ;declare label 1
1DECLARE(L2)         ;declare label 2
1DECLARE(L3)         ;declare label 3
1LSEL1(0,2,1)        ;define inputs and code

1A20                 ;set acceleration
1V5                  ;set velocity
1O(000)              ;set all outputs low
1END

1L1:                 ;label 1 code
1O(1)                ;set output 1 high
1D1000               ;set distance to 1000 steps
1G                   ;move 1000 steps
1T1                  ;wait for 1 second
1O(0)                ;set output 1 low
1END

1L2:                 ;label 2 code
1O(01)               ;set output 2 high
1D-2000              ;set distance to -2000 steps
1G                   ;move -2000 steps
1T1                  ;wait for 1 second
1O(00)               ;set output 2 low
1END

1L3:                 ;label 3 code
1O(001)              ;set output 3 high
1D3000               ;set distance to 3000 steps
1G                   ;move 3000 steps
1T1                  ;wait for 1 second
1O(000)              ;set output 3 low
1END
```

Note: The routine will only run when it receives a valid input pattern corresponding to the numbered label names.

Upon receipt of a valid numeric input pattern the controller runs the associated routine. For example, binary pattern 3 causes routine L3 to run. This routine must finish (reach the END command) before the inputs can be automatically scanned again. The state of the inputs is presented to the controller as a parallel bit pattern. Invalid binary patterns (for non-existent labels) are ignored.

When using the label selection function you must be aware that altering any basic operating parameters, such as velocity, in a routine will change the value used in subsequent routines. Consequently, you will need to define fully the move required in each subroutine block. This can be arranged by the USE command.

System Variables

System variables are named variables held within the drive's controller that are used for storing a variety of system values and settings. Read system variables using the Report system parameter (R command), but note, you can only write to certain variables using the Write (W command).

Certain system variable values may be tested using the IF command. This allows conditional branching within the program code, enabling equal to, not equal to, greater than or less than decisions to be made. Wait for trigger (TR command) can also test certain system variables by delaying code execution until the value of a system variable matches some stored number or string within the program. Refer to the later section on conditional code.

Table of System Variables

Table 4-2 lists system variables in alphabetic order together with their read/write status and range of values stored.

Var	Name	R	W	Range/default value
AB	Analogue Deadband	Y	Y	0 to +255, default = 0
AI	Analogue Input	Y	N	-2047 to +2047
AO	Analogue Offset	Y	Y	-2047 to +2047, default = 0
BR	BAUD rate	Y	Y	9600 or 19200 bits per second (9600 default)
BU	Buffer usage	Y	N	0 to 100% of program buffer used
CQ	Command queuing	Y	Y	1= Pauses until move complete (default) 0= continuous execution
DC	Damping Configuration	Y	Y	0 = settling time damping OFF (default) 1 = settling time damping ON
DF	Drive Fault status	Y	N	See below:
DF1	Drive Fault status	Y	N	First byte of 32-bit DF variable
DF2	Drive Fault status	Y	N	Second byte of 32-bit DF variable
DF3	Drive Fault status	Y	N	Third byte of 32-bit DF variable
DF4	Drive Fault status	Y	N	Fourth byte of 32-bit DF variable
EI	Encoder Input	Y	Y	0=step/dir, 1=cw/ccw, 2=quad ABZ, de-energise drive to change
EM	Encoder count per rev.	Y	Y	1 to 4200000 (default 4000)
EO	Encoder signal Output	Y	Y	0=step/dir, 1=cw/ccw, 2=quad ABZ, de-energise drive to change
EQ	Echo Queuing	Y	Y	0=normal, 1=wait for <CR>, 2=cmd response only
ES	Energise Sense	Y	Y	Sets the sense of the external enable/shutdown_bar signal 0=low signal to enable 1=high signal to enable
EX	Comms. Response Style & Echo Control & Physical Interface (RS232)	Y	Y	0= speak when spoken to, echo off, default for RS485 1= speak whenever, echo off 2= speak when spoken to, echo on 3= speak whenever, echo on, default for RS232

Table 4-2. List of System Variables

Var	Name	R	W	Range/default value
FB	Fieldbus Baud			Refer to CANopen user guide
FC	Fieldbus Control			Refer to CANopen user guide
FN	Fieldbus Node ID			Refer to CANopen user guide
FP	Fieldbus Protocol	Y	Y	Refer to CANopen user guide
HF	Home Final velocity	Y	Y	Sets the final velocity of the home move Range: 0.001 to 5.0 rps (default 0.1)
IC	Input/Output Configuration	Y	Y	Input pull-up/down, output source/sink configuration 0 to 8191 default:8160
IN	Inputs (on drive)	N	N	Local drive inputs 1 to 5, same format as IS command
INn	Inputs (expansion)	N	N	Fieldbus expansion inputs, IN1=bank1, IN2=bank2.
IP	In Position flag	Y	N	1= In position or 0= not yet in position
IT	In Position Time	Y	Y	1 to 500mS, default=10mS
MS	Motor Standby	Y	Y	Range 10% to 100% of programmed current (default 50%)
MV	Moving	Y	N	Flag 1= moving or 0 = not moving
PA	Position Actual	Y	N*	-2,147,483,648 to 0 to 2,147,483,647
PE	Position Error	Y	N*	+/- 65535
PF	Position Following	Y	Y	-2,147,483,648 to 0 to 2,147,483,647
PI	Position Incremental	Y	Y	-2,147,483,648 to 0 to 2,147,483,647
PM	Position Master	Y	Y	-2,147,483,648 to 0 to 2,147,483,647 Note: a write to PM sets the modulus
PR	Position Registration	Y	N	The primary (X2) feedback position (PA) on the last active transition on input 2 (start of valid REG move). Range: -2,147,483,648 to 0 to 2,147,483,647
PS	Position Secondary	Y	N	The PM count position on the last active transition on input 1 (falling edge viewed using IS). Range: -2,147,483,648 to 0 to 2,147,483,647
PT	Position Target	Y	Y	-2,147,483,648 to 0 to 2,147,483,647 Trajectory generator open loop target position
RB	Ready/Busy flag	Y	N	Flag 0= ready or 1= busy
RM	Registration Move	Y	N	Flag 1= reg move in progress 0 = not doing reg move
RV	ReVision of software	Y	N	x.yy major.minor
SC	S Curve configuration	Y	Y	0 = S curve accel/decel disabled (default) 1 = S curve accel/decel enabled
SN	Serial number	Y	N	reserved

Table 4-2. List of System Variables (Continued)

Var	Name	R	W	Range/default value
ST	Status of indexing	Y	N	See below
ST1	Status of indexing	Y	N	First byte of 32-bit ST variable
ST2	Status of indexing	Y	N	Second byte of 32-bit ST variable
ST3	Status of indexing	Y	N	Third byte of 32-bit ST variable
ST4	Status of indexing	Y	N	Fourth byte of 32-bit ST variable
TT	Trigger Timeout	Y	Y	Optional timeout for trigger command 0-65 seconds in 0.01 increments. User status bit 8 is set to indicate timeout occurred before trigger condition met. Bit is clear if trigger condition met before timeout. The default time is = 0.00 (no timeout).
UF	User program Fault status	Y	N	See below
UF1	User Fault Status	Y	N	First byte of 32-bit User Fault status word
UF2	User Fault Status	Y	N	Second byte of 32-bit User Fault status word
UF3	User Fault Status	Y	N	Third byte of 32-bit User Fault status word
UF4	User Fault Status	Y	N	Fourth byte of 32-bit User Fault status word

*Can be set to 0 only.

Table 4-2. List of System Variables (Continued)

AB, AI and AO Description

AB controls the dead band and AO the offset of the differential analogue speed control input. See ***Differential Analogue Input*** in the ***Electrical Installation*** section.

BR Description

This sets the Baud rate of serial communications. Enter the required Baud rate directly, for example **aW(BR,19200)** to set the rate to 19200. You will need to save this setting and then reset the drive (Z command) or cycle the power before the change will take effect.

BU Description

Gives the total percentage of program buffer usage, unlike an **aDECLARE** that gives the percentage of buffer room for each label, subroutine.

CQ Command Queuing

Enable command queuing in mode incremental/absolute to buffer each command waiting for the previous command to complete, before issuing the next. In certain circumstances, disable this sequential operation, for example if you need to generate a trigger pulse part way through a move. Normally, the move would complete before trigger command execution, but by disabling command queuing, the trigger command becomes immediate and will operate upon meeting the required trigger conditions.
For example, the following code would allow output 1 to signal PA is greater than 10000 before finishing the move.

```
1MAIN:              ;define label
1MI                 ;mode incremental
1W(CQ,0)            ;enable continuous execution of commands
1G                  ;go
1TR(PA,>,10000)     ;trigger when position actual becomes greater than 10000
1O(1)               ;output 1
1TR(IP,=,1)         ;wait for move to finish
1W(CQ,1)            ;enable command queuing again
1END
```

DC Damping Configuration

Selecting DC gives a faster settling time by damping oscillations (ringing) of the motor shaft. Under certain conditions, such as use with low current motors, the activation of the damping circuit can lead to an increase in the audible noise of operation. However, we recommend the use of DC for highly dynamic operations.

DF Description

See drive fault bit description in ***Reporting the Status of Variables***.

EO Description

When an encoder is connected to the primary feedback input on X2, you may use the encoder outputs (connector X4) to supply a step-direction or step-up/step-down signal for use by another drive. System parameter EO determines the output as defined in Table 4-3. Note: The source of these pulses is X2 primary encoder, they are not generated from within the drive's indexer. Before changing the system variable EO it is necessary to de-energise the drive.

X4	EO=0	EO=1	EO=2
14	STEP+	CW+	A+
9	STEP-	CW-	A-
15	DIR+	CCW+	B+
10	DIR-	CCW-	B-

Table 4-3. Encoder Output Configuration

EI Description

System parameter EI, controls encoder inputs (connector X4) as defined in Table 4-4.

X4	EI=0	EI=1	EI=2
12	STEP+	CW+	A+
7	STEP-	CW-	A-
13	DIR+	CCW+	B+
8	DIR-	CCW-	B-

Table 4-4. Encoder Input Configuration

CAUTION

De-energise the drive before changing EI and EO.

EQ Description

Echo queuing (EQ) is a system variable that can be useful for multi-axis control programs where you need to send and receive messages from individual drives controlled from a PC. The variable controls the way messages are echoed and its use prevents corruption of commands by system response messages. In a normal multi-axis system, commands from the main controller are, in turn, echoed from drive to drive throughout the system and can be finally returned to the main controller. If a command is transmitted whilst a drive is supplying a response the two messages will interact, effectively destroying one another. Setting EQ to mode 1 prevents a drive from issuing a response until it receives a carriage return, thereby delaying its response until it finishes receiving. This stops the corruption of messages, which can now be read back in a complete form.

EQ can only be used with a report or write command, as follows:

R(EQ) reads the current setting of the system variable.

W(EQ, 0 - 2) sets the EQ system variable to operate in mode 0, 1 or 2.

Mode 0 sets the standard operating mode where characters are echoed as they are sent.

Mode 1 does not allow any characters to be echoed until a carriage return is sent. This prevents complete messages from being split if a data collision occurs.

Mode 2 allows only the response from a command to be sent, not the command itself. This minimises the amount of data being transferred and therefore helps to reduce the chance of a transmit buffer overflow.

Note: The set address command (#) will be echoed irrespective of the state of the echo queuing variable.

ES Description

System variable ES controls the required polarity of signal on the enable/shutdown_bar input (X4 pin 11). The default value of ES is zero (ES=0), therefore to enable the drive connect X4 pin 11 to X4 pin 4 (0V). With ES=1 X4 pin11 may be left open circuit to enable the drive. To energise the drive, the drive must be enabled and the ON command issued. The function of this input differs when in mode 'MP', please refer to the ***Command Reference*** section for more details.

EX Description

System variable EX controls the style and protocol of the drive's serial communications link.

IP, IT and MV Description

System flag variables IP (In Position) and MV (Moving) together with variable IT (In position Time) interact with one another as shown in Figure 4-2. The MV flag is only high whilst commanded motion is taking place. The IP flag can only go high once movement has stopped and the IT timer value has timed-out. Consequently you need to set IT to a time long enough to ensure velocity variations (ringing) has ceased.

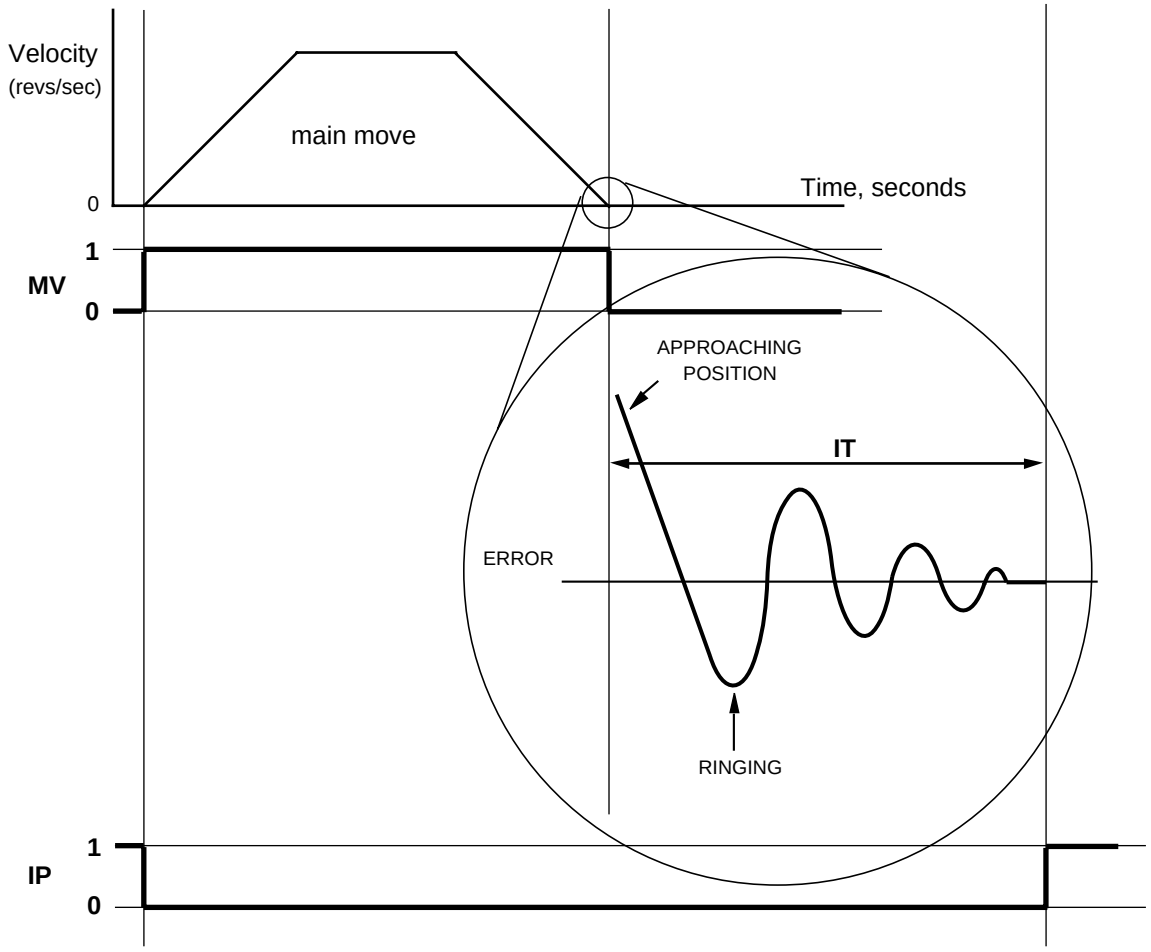


Figure 4-2. Interaction of MV, IP, & IT

You do not have to wait for the IP flag to be set at the end of every move, but its use improves positioning accuracy.

Example Use this code after each G command to improve positioning.

```
1MI          ; mode incemental
1W(CQ,0)     ; turn off command queuing
1G           ; start the move
1T0.1        ; wait 100ms
1IF(MV,=,1)
1"Moving"
1TR(IP,=,1)
1"Stopped"
1W(CQ,1)     ; re enable command queuing
```

HF Description

HF sets the final home velocity when you perform a GH command.

IC Description

See ***IC System Variable*** in the ***Electrical Installation section***.

IN Description

The IN system variable is equivalent to the IS command, but allows individual inputs to be tested using IF and TR commands during conditional coding.
For example:

The following test looks for input 1 low and input 3 high.

```
IF(IN,=,0X1XX)
```

Where X=don't care.

INn Description

The INn system variable is used to define a particular bank of inputs when used with Fieldbus input expansion modules.

MS Description

When the motor is stationary, reduce its current to minimise heating or to conserve power. MS sets the reduction in current as a percentage of the programmed current (the value set in the MOTOR command). When selected, the drive will switch to standby 25mS after the last motor step.

Motor standby current reduction is capped at a value of 70% of the drive's maximum output current. Consequently, if you attempt to set an MS value greater than 70 the current reduction value will always be equal to 70% of the drive's maximum output current. For example, using a ViX500 (max. output current of 5.6A) and setting MS to 90 will give a current reduction value of 4A (70% of 5.6A).

PA Description

PA reports the actual position of the motor shaft, assuming a primary encoder is fitted. Although PA is marked as being read only it will accept the value 0 to be written to it for resetting purposes. If you perform a **W(PA,0)** system variables PF, PE and PT will also be set to 0.

PE Description

PE reports the position error, that is, the difference between PT and PA.

PF Description

PF reports the position fed-back by a remotely mounted encoder for following applications. This is the position demanded by the following input. Counts are only recorded when following is enabled and at the scaled rate, this means if the scale is -50% and 4000 counts are received by the drive, PF will read –2000.

PI Description

PI reports the distance moved by the last move (G) command.

PM Description

PM reports the number of counts received from power-on by the following input. No scaling is applied and PM counts regardless of following being on or off. Writing a number to PM sets the modulus for count wrapping. That is, writing a specific number of counts to PM sets the count required before the drive re-starts counting from zero again. This is useful if you wish to know the position of the motor shaft as an arbitrary count.

For example writing a count of 4000 to PM means that for every shaft rotation a new count of 0 to 3999 is started (until the absolute count limit is reached). By reading PM, a count will be returned that is somewhere between 0 and 3999, the exact value being an indication of the instantaneous shaft position.

PR Position Registration Description

PR always reports the position of the motor from the primary feedback (X2 connector) signal on the last active transition on user input 2. The signal is only active at the start of a valid REG move.

PS Position Secondary Description

PS reports the position of the following input from the secondary feedback (X4 connector) signal on the last active transition on user input 1.

PT Description

PT reports the open loop target position of the motor, that is, where you have commanded the motor to move to.

RB Description

Reports the state of the controller as being ready or busy. While executing a program or subroutine the controller is busy.

RM Description

Reports a 1 if a registration move is being actioned.

RV Description

Reports the revision of software being used by the controller.

SC S-Curve configuration

To reduce the amount of *jerk* (rate of change of acceleration or deceleration) within a move, enable SC. When enabled, this variable smoothes-out rapid changes of acceleration, as shown in Figure 4-3.

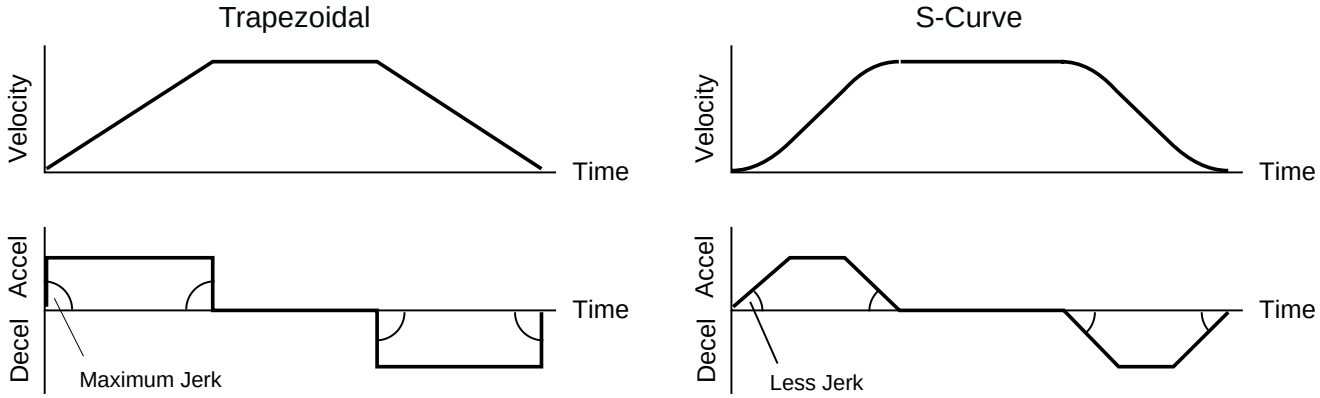


Figure 4-3. S Curve Correction of Moves

To achieve this type of S curve correction an average acceleration value is used which is set at half the value of the maximum acceleration. In all cases, the value of AA will be used for acceleration and deceleration. If a value of AD is set that is not equal to AA, then the value of AA will be used for all acceleration and deceleration settings. Asymmetric move profiles are not possible when using S-curve correction.

Since the peak acceleration will be twice that of AA, this needs taking account of when performing any torque calculations.

SN Description

reserved.

ST Description

See reporting of status bits in ***Reporting the Status of Variables***.

TT Description

The trigger timeout can be set or read using TT. If a timeout occurs status bit 8 is set high. Note: Setting a value of 0.00 results in NO trigger timeout.

Example:

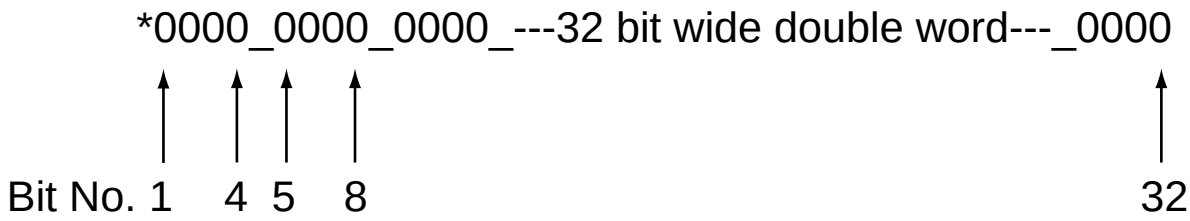
```
1W(TT,1.5)          ;timeout after 1.5 seconds
1G                  ;do the move
1TR(IN,=,1)         ;wait for input 1 to activate or timeout
1IF(ST1,=,XXXXXXX1) ;check for timeout
1GOTO(TOUT)         ;jump to 'TOUT' routine
1"IN1 ON"           ;else display message over comms. link
.                   ;continue code
```

Reporting the Status of Variables

By examining Table 4-2 you can see that most system variables take a numerical value or record a simple ON/OFF state (0 or 1 Flags). Certain variables perform a reporting function which provides you with information on the status of the indexer and any drive faults present in the hardware or user program code.

Status Variable Reporting

Variable ST is a 32-bit double word that contains status information.
When read, ST reports a 32-bit double word pattern of the form:



Where a bit is set (displayed as a 1) its bit number can be determined and compared with the bit number value given in Table 4-5 to determine the Status Information being reported. Use the Read command to display the ST word pattern, that is '**aR(ST)'.**

Bit Number	Bit Tested	Status Information
1	ST1.1	Command processing paused
2	ST1.2	Looping (command executing)
3	ST1.3	Wait for trigger (input)
4	ST1.4	Running program
5	ST1.5	Going home
6	ST1.6	Waiting for delay timeout
7	ST1.7	Registration in progress
8	ST1.8	Last trigger command timed out
9	ST2.1	Motor energised
11	ST2.3	Event triggered - active until trigger inputs are reset
12	ST2.4	Input in LSEL not matching label
13	ST2.5	-ve limit seen during last move
14	ST2.6	+ve limit seen during last move
16	ST2.8	Reserved
17	ST3.1	Executing a position maintenance move
18	ST3.2	Possible stall
19	ST3.3	Moving (in motion)
20	ST3.4	Stationary (in position)
21	ST3.5	No registration signal seen in registration window
22	ST3.6	Cannot stop within the defined registration distance
23	ST3.7	Reserved
24	ST3.8	Reserved
25	ST4.1	In motion, 0 for positive motion, 1 for negative motion
26	ST4.2	Reserved
27	ST4.3	Following enabled = 1, not following = 0
28	ST4.4	STOP input active
29	ST4.5	Load mounted encoder enabled
30	ST4.6	Scaling enabled
31	ST4.7	Command input inverted

Table 4-5. Status Bits Description

Status Variable Byte Reporting

A convenient and more compact way of interrogating the status variable is to test it a byte at a time using the **STn** within a read command, where n is used to select the byte to be tested. For example to read or test the first 8 bits (first byte) of the ST variable status word, use ST1. Since the status word consists of 4 bytes the relevant part of the word can be read using ST1 (bits 1 to 8), ST2 (bits 9 to 16), ST3 (bits 17 to 24) or ST4 (bits 25 to 32).

Fault Status Reporting

Faults are classified into two groups:

Drive Faults DF (hardware faults present in the drive)

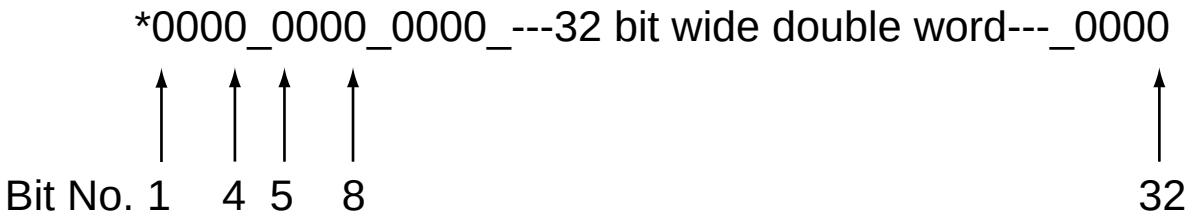
or

User Faults UF (user program faults)

Drive Faults

Hardware drive faults will cause the drive output stage to be turned OFF (de-energised). This will cause the Drive LED to turn RED. Once the fault has been corrected the drive may be re-energised using the ON command.

When read, DF reports a 32-bit double word pattern of the form:



Where a bit is set (displayed as a 1) its bit number can be determined and compared with the bit number value given in Table 4-6 to determine the Drive Fault being reported.
Use the Read command to display the DF word pattern, that is '**aR(DF)'.**

Bit	Bit Tested	Stop	Type	DF Information
1	DF 1.1			Composite fault
2	DF 1.2	K	T	+/-15V supply rail
3	DF 1.3	K	R	Motor HV under-voltage trip point reached
4	DF 1.4	K	R	Motor HV over-voltage trip point reached
5	DF 1.5			
6	DF 1.6	CD	R	Vio over-voltage trip point reached
7	DF 1.7	K	T	Encoder / Auxiliary 5V under voltage trip
8	DF 1.8	K	SLEEP	Impending power loss, V I/O under voltage (24V – logic supply)
9	DF 2.1			Reserved
10	DF 2.2			Reserved
11	DF 2.3	CD	R	Motor over temperature
12	DF 2.4	CD	R	Ambient over temperature
13	DF 2.5	CD	R	Drive over temperature
14	DF 2.6	K	T	Incompatible firmware version
15	DF 2.7	K	T	Unrecognised power stage
16	DF 2.8	K	T	Controller diagnostic failure
17	DF 3.1	K	R	Output stage over current
18	DF 3.2	CD	R	Output driver over current
19	DF 3.3	C	R	Tracking limit exceeded : Stall condition
20	DF 3.4			Reserved
21	DF 3.5	CD	R	Drive disabled – check enable input and state of ES variable
22-24	DF 3.6/8			Reserved
25	DF 4.1	K	T	Watchdog 1
26-31	DF 4.4/7			Reserved
32	DF 4.8			CAN I/O errors

Key:
C : Performs controlled stop.
CD : Controlled stop then de-energise
K : Performs motion kill – quick stop. Possible instant de-energise depending on fault source.
R : Recoverable without power cycle
SLEEP : Drive shuts down completely – no comms, requires power-cycle to recover
T : Terminal (requires power cycle or repair before drive will energise / operate once again)

Table 4-6. Drive Fault Bit Description

See **Maintenance & Troubleshooting** for a more detailed explanation of Drive Faults.

Drive Fault Byte Reporting

In exactly the same way as the status variable, the drive fault status can be reported a byte at a time, using **DFn** within a read command.

User Faults

User faults can be caused by programming errors, such as issuing a GO command when the drive is de-energised. They are reported in a 32-bit word format the same as Drive Faults.

Performing a read UF command will report the current state of any User Faults listed in Table 4-7.

Bit Number	Bit Tested	UF Information
1	UF 1.1	Value is out of range
2	UF 1.2	Incorrect command syntax
3	UF 1.3	Last label already in use
4	UF 1.4	Label of this name not defined
5	UF 1.5	Missing Z pulse when homing
6	UF 1.6	Homing failed - no signal detected
7	UF 1.7	Home signal too narrow
8	UF 1.8	Drive de-energised
9	UF 2.1	Cannot relate END statement to a label
10	UF 2.2	Program memory buffer full*
11	UF 2.3	No more motion profiles available
12	UF 2.4	No more sequence labels available
13	UF 2.5	End of travel limit hit
14	UF 2.6	Still moving
15	UF 2.7	Deceleration error
16	UF 2.8	Transmit buffer overflow
17	UF 3.1	User program nesting overflow
18	UF 3.2	Cannot use an undefined profile
19	UF 3.3	Drive not ready
22	UF 3.6	Save error
23	UF 3.7	Command not supported by this product
24	UF 3.8	Fieldbus error
25	UF 4.1	Input buffer overflow
26	UF 4.2	Reserved
27	UF 4.3	Command not actioned
28	UF 4.4	Scale distance is non-integer
29 to 32	UF 4.5/8	Reserved

Table 4-7. User Fault Bit Description

*sends an ASCII ‘bell’ character to indicate a buffer overflow condition.

User Fault Byte Reporting

In exactly the same way as the status variable, the user fault status can be reported a byte at a time, using **UFn** within a read command. For example to read or test the first 8 bits (first byte) of the UF variable status word, use UF1. Since the status word consists of 4 bytes the relevant part of the word can be read using UF1 (bits 1 to 8), UF2 (bits 9 to 16), UF3 (bits 17 to 24) or UF4 (bits 25 to 32).

Resetting User Fault Bits

The User Fault variable (UF) is cleared to all zeroes once it has been read by issuing a **R(UF)** command. Reading individual bytes of the User Faults variable will not clear any particular byte, so issuing a **R(UF2)** command will keep byte 2 bits intact. Also testing a particular byte using the **IF** or **TR** command will keep bits intact.

Note: sending the drive an **ON** command will immediately clear the User Fault variable, all bytes will be set to 00000000.

Byte Testing

Remember, the code can be used to test a particular byte of the User Fault word. For example:

```
1IF(UF2,<>,10X10X10)    ; if contents of UF2 does not equal 10X10X10 execute
                        ; the next line of code, otherwise skip the next line
1A500                   ; acceleration and deceleration changed to 500rps2 if
                        ; previous test was true
1R(UF2)                 ; read the value of byte 2 of the user fault status word
*01010101               ; contents of byte 2
```

Note: When UF2 is tested or read it is not cleared to all zeroes.

This example uses a conditional test to compare UF2 with 10X10X10. The use of conditional tests within IF and TR commands is described in the ***Conditional Code*** sub-section.

Reporting System Information During Code Development

Whilst developing a program using EASI-Tools, it is likely that certain blocks of code when downloaded to the drive will return an **E* error code. To analyse the cause of the error you can make use of EASI-Tools Status report window which, when read, will report back the cause of the error. For example, selecting status report 'User' following a **E* may report back 'Label of this name not defined'.

Within EASI-Tools a system variable can be read using the status report window or using the report command directly from the terminal window (For example **3R(ST)**). Using this style of report an immediate response will be returned which will not be saved within the program code. If you wish to save the response, use the single byte version of the report command, that is **3R(ST1)**, **3R(ST2)**, **3R(ST3)** or **3R(ST4)** depending upon which byte of the variable you wish to capture.

If the indexer is waiting on a trigger command, you can still send an interrogation command such as 1R(RB), 1R(DF1), 1R(ST1), 1R(UF1), 1IS, 1O, 1A .....and a report will be returned. However, if a buffered command is sent, such as G or 1A10, then all future interrogation commands are buffered, apart from 1R(RB), 1R(DF), 1R(ST) and 1R(UF).

Conditional Code

The flow of a motion control program will depend upon the position of the motor in combination with the value of particular inputs and commands. System variables are used to continuously monitor the state of a drive's indexer and are able to report such things as 'status of indexing' or 'moving'/not moving' as listed in Table 4-2. Certain system variables are capable of being tested by the TR (wait for trigger) or IF (test condition) commands. This allows the value of a system variable to be tested in the following ways:

=	Equals
<>	Does not equal
>	Greater than
<	Less than

The TR command pauses program execution until the required trigger condition is met, while the IF command tests the value of a system variable and executes the next line of code if it is true, otherwise it skips the next line of code. Use of these commands allows synchronisation with external events and program branching.

System variables which may be used in conjunction with the IF command are listed in Table 4-8. Where the variable can also be used with the TR command a 'Y' appears in the TR column.

Variable	Name	>	<	=	<>	TR	Format
AI	Analogue input	Y	Y	N	N	Y	decimal
DFn	Drive fault status	N	N	Y	Y	Y	binary
IN	Inputs (drive)	N	N	Y	Y	Y	binary
INn	Inputs (expansion)	N	N	Y	Y	Y	binary
IP	In position flag	N	N	Y	Y	Y	bit
MV	Moving	N	N	Y	Y	Y	bit
PA	Position absolute	Y	Y	Y*	Y	Y	decimal
PE	Position error	Y	Y	Y*	Y	Y	decimal
PF	Position following	Y	Y	Y*	Y		decimal
PI	Position incremental	Y	Y	Y*	Y	Y	decimal
PM	Position master	Y	Y	Y*	Y	Y	decimal
PT	Position target	Y	Y	Y*	Y	Y	decimal
RM	Registration move	N	N	Y	Y	N	bit
STn	Status of indexing	N	N	Y	Y	Y	binary
UFn	User program fault status	N	N	Y	Y	N	binary

* Not recommended during motion

Table 4-8. System Variables that can be used for Conditional Control

Conditional Code Example

The following code is a good example of how the conditional IF statement can be used for fault diagnosis within the FAULT label.

```
1FAULT:                          ;define check label
        1IF(UF2,=,XXXXXX1X)      ;deceleration error
            1"Decel_Err"
        1IF(DF1,<>,00000000)     ;warning of a drive fault
            1"Drive_Flt"
        1IF(ST1,=,XXXXX1XX)      ;waiting for a delay timeout
            1"Delay_tout"
        1IF(ST2,=,1XXXXXXX)      ;motor is energised
            1"Motor_On"
        1T1                      ;wait 1 second
1END                             ;end of definition
```

Command Queuing

Command queuing in mode incremental is normally enabled, this means commands are buffered, each command waiting for the previous command to complete before the next one is issued. In certain circumstances this sequential operation needs to be disabled, for example if you need to generate a trigger pulse part way through a move. Normally, the move would complete before the trigger command is executed, but by disabling command queuing the trigger command becomes immediate and will operate when the required trigger conditions are met.
For example, the following code would allow output 1 to signal PA is greater than 10000 before finishing the move.

```
1MAIN:              ;define label
1W(CQ,0)            ;enable continuous execution of commands
1G                  ;go
1TR(PA,>,10000)     ;trigger when position absolute becomes greater than 10000
1O(1)               ;output 1
1W(CQ,1)            ;enable command queuing again
1END
```

Motion Control Using the EASI Command Set

Move Types

Mechanical movement results from the motion of a motor shaft. By controlling the velocity, acceleration, distance and direction of the motor, different move profiles can be created for particular applications. Move types can be preset, meaning a move is made in a controlled way over a specified distance, or continuous where only acceleration, velocity and direction are defined, distance being ignored. Various move types can be selected using the mode (M) command.

Preset Moves

Preset moves allow you to position a target or work-piece in relation to the motor's previous stopped position (incremental moves) or in relation to a defined zero reference position (absolute moves).

Absolute Preset Moves (MA)

An absolute preset move will move the shaft of the motor a specified distance from the absolute zero position (MA).

Incremental Preset Moves (MI)

When the MODE command is used to select indexed move with incremental positioning (MI), the motor shaft can be moved a specified distance from its starting position in either a clockwise (CW) or counter clockwise (CCW) direction.

Note: a positive direction is defined as one resulting in clockwise (CW) rotation of the motor shaft when viewed from the flange.

Continuous Moves (MC)

This mode is useful for applications which require constant travel of the load, when the motor must stop after a period of time has elapsed rather than after a fixed distance, or when the motor must be synchronised to external events such as trigger input signals (MC).

Motor Direction & Positive Motion

A positive direction command usually produces clockwise (CW) rotation of the motor shaft when viewed from the shaft end*.
However, when limit switches are used it is important that the positive direction produces motion towards the positive limit switch (see sub-section on HOMING). If this is not the case, interchange the motor connections to A+ and A- to reverse motor direction.

* In practice this depends on internal motor wiring which varies between motor manufacturers.

Motion Profiles

In any motion control application the most important requirement is precise, controlled shaft rotation, whether it be with respect to position, time or velocity. This pattern of movement is called a Motion Profile. Generally, such a profile can be represented graphically in the form of a diagram of time or distance moved plotted against velocity. For example, the triangular shaped profile shown in Figure 4-4 would be obtained if you programmed either a very low acceleration or a very high velocity or both over a relatively short distance.

Triangular Profile

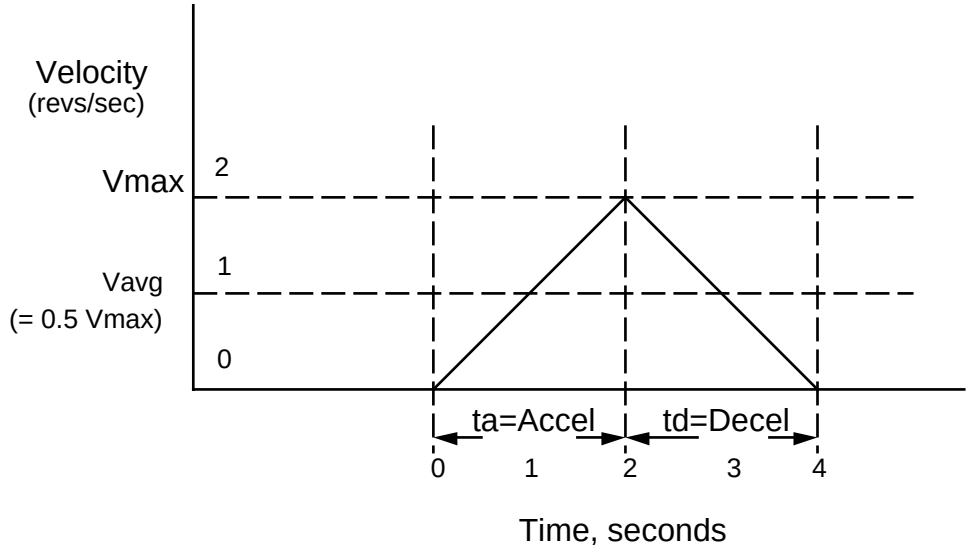


Figure 4-4. Triangular Profile

Setting the acceleration to 1 rev/sec^2 with the velocity set to 5 revs/sec over a distance of 16000 steps (4 revs), a triangular motion profile will result. This is because by the time the motor shaft has reached a velocity of 2 revs/sec, it will also have travelled half of the defined distance due to the acceleration setting of 1 rev/sec^2.

Trapezoidal Profile

A trapezoidal move profile results when the defined velocity, you have programmed, is attained before the motor shaft has moved half of the specified distance. This is due to a defined velocity that is low, a defined acceleration that is high, a move distance that is long, or a combination of all three. For example, if the acceleration is set to 10 revs/ sec^2, velocity is set to 1 rev/sec, and distance is specified as 20000 steps (5 revs), the resulting motion profile would look like this:

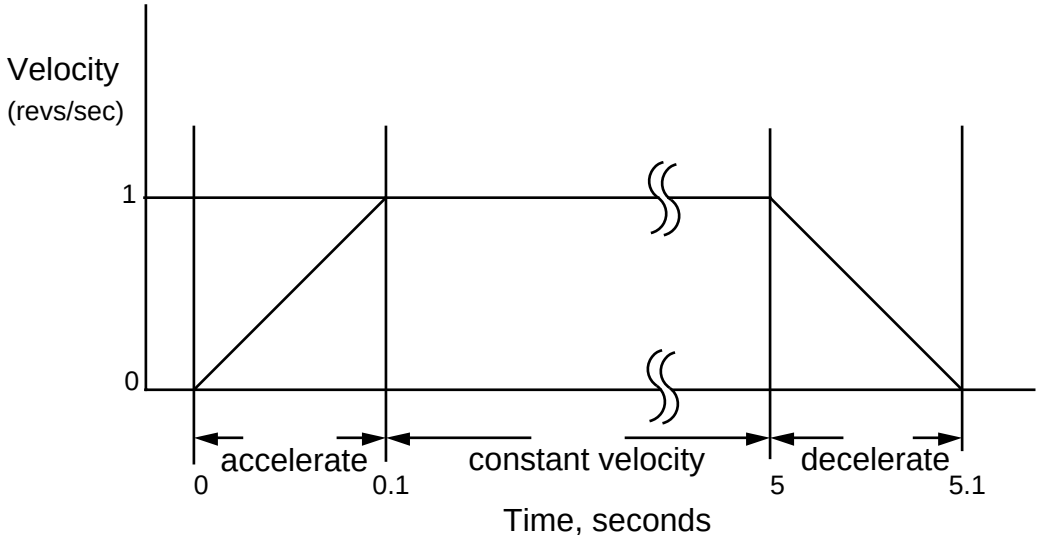


Figure 4-5. Trapezoidal Profile

Registration

One of the major uses of registration is for packaging and labelling applications where a registration mark or label edge is used to sense the position or orientation of an object. Once detected a registration move can be triggered, which is a separate independent move that, for example, may position a jar for a labelling operation. The registration move itself often needs to be performed quickly (faster than the current move, to prevent queuing in serial batch processes), Figure 4-6 illustrates a typical registration move.

Note: A registration move is always performed in mode incremental, even if the drive is configured for mode absolute

The REG command once turned ON (**1REG1**), defines a registration move which can be superimposed upon a standard move profile. The registration move will only be performed if a specified input edge is detected on the registration input. If an optional hold-off distance has been defined the registration command will only respond to a registration input occurring beyond the hold-off distance. Otherwise, once the basic move had started, any valid registration input or mark would trigger a registration move immediately. Also, if an optional registration window has been defined, a registration move can only be triggered if the registration mark occurs within the registration window.

Once a valid registration mark has been detected the registration move is performed using the move parameters taken from the previously defined profile* (profile_number in the command parameters). At the end of the registration move the user program GOSUBs to the code immediately following the REG label. If no registration mark is detected, the standard move profile completes and the user program GOSUBs to the code immediately following the NOREG label.

* Registration will always occur in the current move direction. If the direction in the defined profile is different to the current move direction, the direction information in the defined profile is ignored.

An optional output can be programmed to indicate that a move that has been armed is ready for registration. This would normally be after the move has started or after the hold-off distance (if defined). The output chosen must be within the range of allowable outputs (0 to 3). The default value is 0 (no output).

If the REG move must immediately begin to decelerate to achieve the distance programmed, the REG profile is not configured correctly and the deceleration rate used will not be the requested rate. In this case, the registration move may appear to be performed, but the NOREG label is executed.

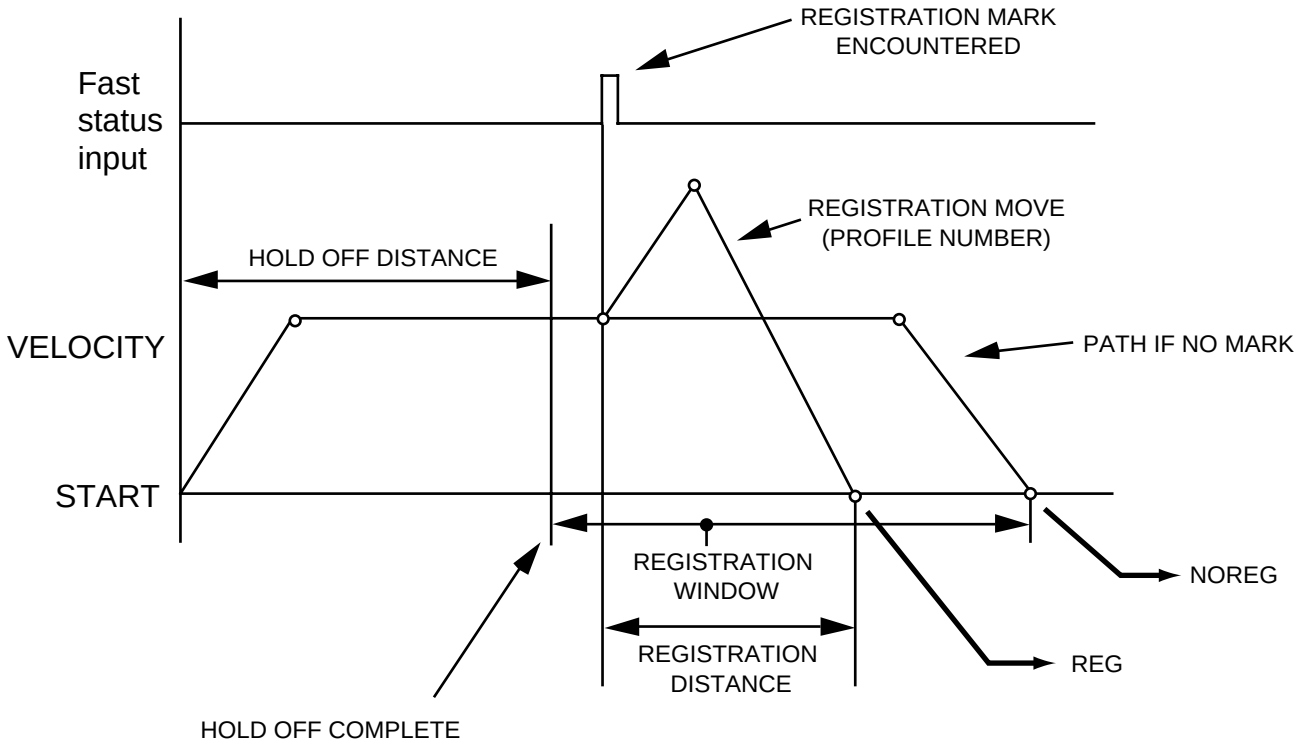


Figure 4-6. Registration Move Profile

A successful registration will cause the code, following the registration move, to jump to the REG label, from which normal program operation can continue.

Before you can perform a registration move, the following code elements must be in place:

1. Enable the registration function.
2. Completely specify the registration move required, in terms of distance, velocity, acceleration and deceleration.

Once a registration move has been defined, registration can be enabled/disabled using **aREG1** (to turn it ON) or **aREG0** (to turn it OFF), where 'a' defines the axis address.

When registration is enabled, any valid input edge will activate the registration move (whilst moving), however once activated any subsequent edge will have no effect. Consequently once the registration signal has been accepted for the current move all other registration signals will be ignored until a new move has been started.

An example of registration code is given below:

```
1START:                 ;start label definition
1PROFILE4(10,10,40000,5)  ;define move parameters
1PROFILE5(20,20,10000,10) ;define move parameters
1REG1(1,5,5000)         ;define registration move parameters & arm registration

1USE(4)                 ;use motion profile 4
1G                      ;execute move
1END                    ;end of start label

1REG:                   ;on reg mark valid turn on o/p 3 (batch counter)
1O(XX1)
1T0.5                   ;wait for 500ms delay
1O(XX0)                 ;turn off o/p 3
1END                    ;end of label definition

1NOREG:                 ;if reg mark not valid/seen
1O(X1X)                 ;turn on o/p 2
1T0.25                  ;wait for 250ms delay
1O(X0X)                 ;turn off o/p 2
1END                    ;end of label definition
```

Run the above by typing **1GOTO(START)**

Homing

The term ‘homing’ refers to an automatic return to a mechanical reference position which is usually performed when the system is first powered up. All subsequent moves will then be relative to this reference position. The home position is usually determined by an optical or proximity switch, though a mechanical switch can also be used.

Definition Of Terms

To aid the description of homing operations the following terms are defined:

Positive motion - is motion towards the positive limit
Home switch positive edge - is the edge of the home switch on the positive limit side
Home switch negative edge - is the edge of the home switch on the negative limit side
Home switch operating range - is the distance moved whilst the switch is operated

Four of these terms are illustrated in Figure 4-7.

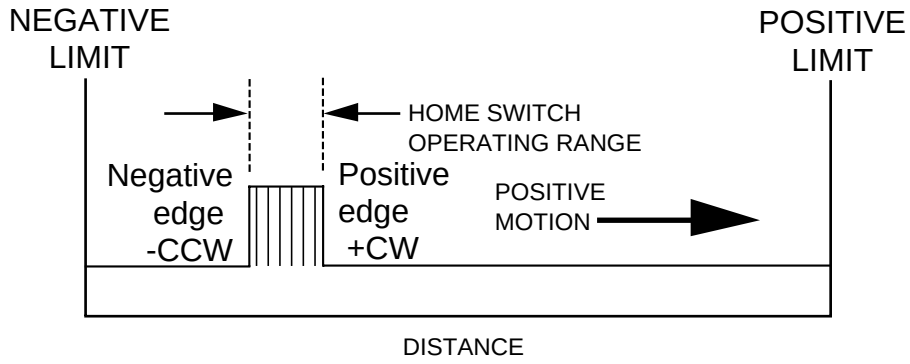


Figure 4-7. Home Switch Operation

Switch Considerations

Any type of switch will have an operating range that may correspond with a significant motion of the motor shaft, depending upon the gear ratio between motor and load. Consequently, just detecting the home switch voltage level will not give a well defined home position. To improve the accuracy it is possible to stop on either the positive or negative edge of the home switch.

Switches generally exhibit a hysteresis characteristic when operated from opposite directions, therefore homing moves always make the final approach to the home switch from the same direction.

Homing Configuration Command

The command allows you to define the mechanical edge of the home switch at which you wish home to be. The command also allows you a choice of home switch type, that is normally closed or normally open, however if you change the switch type this does not change the edge you are homing to. Remember the positive edge is the mechanical edge of the home switch closest to the positive limit.

Other features of the HOME configuration command allow adjustment of the search speed and direction, the acceleration or deceleration rate to be used and mode selection. When setting the deceleration rate you must ensure sufficient distance is left between the home switch and any limit to make sure motion is brought to a halt after the home switch is detected and before a limit is reached. If not, the system will be brought to an immediate halt as soon as the limit is detected.

Mode Selection

Mode selection allows you the choice of how and where motion is brought to a stop within the home switch operating range. The choices are:

- Mode 0 - the indexer will detect the first edge (positive or negative) and will then decelerate to rest within the home switch operating range
- Mode 1 - will cause motion to stop at the mechanical edge of your choice (positive or negative)
- Mode 2 – reserved
- Mode 3 – If an encoder with a Z channel is used then the controller will seek the Z position after detecting the specified home switch edge.
- Mode 4 – If an encoder with a Z channel is used then the controller will seek the Z position without the need for a home switch.

Mode 0 operation simply returns the motor to its home position at some point between the negative edge and positive edge of the home switch. Apart from knowing which edge of the switch was used the exact position within the home switch range is undefined. A more precise home position can be obtained by using mode 1.

Mode 1 allows the home position to be defined as either the positive or negative edge of the home switch. Note, although mode 1 fixes the home position at one of two edges the precise position is still subject to the repeatability of the home switch itself. Practical applications will exhibit variations in switch performance and consequently the home position will still be subject to variation by a small number of motor steps.

Mode 2 Reserved.

Modes 3 & 4 for use with Z channel encoders.

Go Home Command

The go home command (GH) is used to return to the reference home position. Issuing a GH command will cause motion in a direction defined by the HOME configuration command. Figure 4-8 shows the path taken if motion was started between the positive edge of the home switch and the positive limit (positive side of home). The dotted line represents positive movement and the solid line negative, although once past the positive edge of the home switch both merge to follow one common path. Positive movement results in motion towards the positive limit, once the limit is hit motion is reversed* and finally heads for the home switch. Negative motion will immediately head for the home switch.

*Note: Limit inputs must be enabled to allow a move to bounce off a limit.

Assuming home is the positive edge of the home switch, as soon as the edge is detected motion is decelerated to a stop. Direction of travel is reversed and a distance is calculated to move just outside the positive edge of the home switch. This new move is performed in a positive direction. Again motion is stopped, and the direction of travel is reversed and a negative approach is made at a fixed velocity determined by system variable HF. As soon as the positive edge is again detected the motor is stopped.

Note: If the deceleration rate is set too low, the home switch operating range could be travelled through before motion is brought to a stop. If this happens, a warning ‘home switch too narrow’ will be reported, but homing will continue from the other side of the home switch operating range.

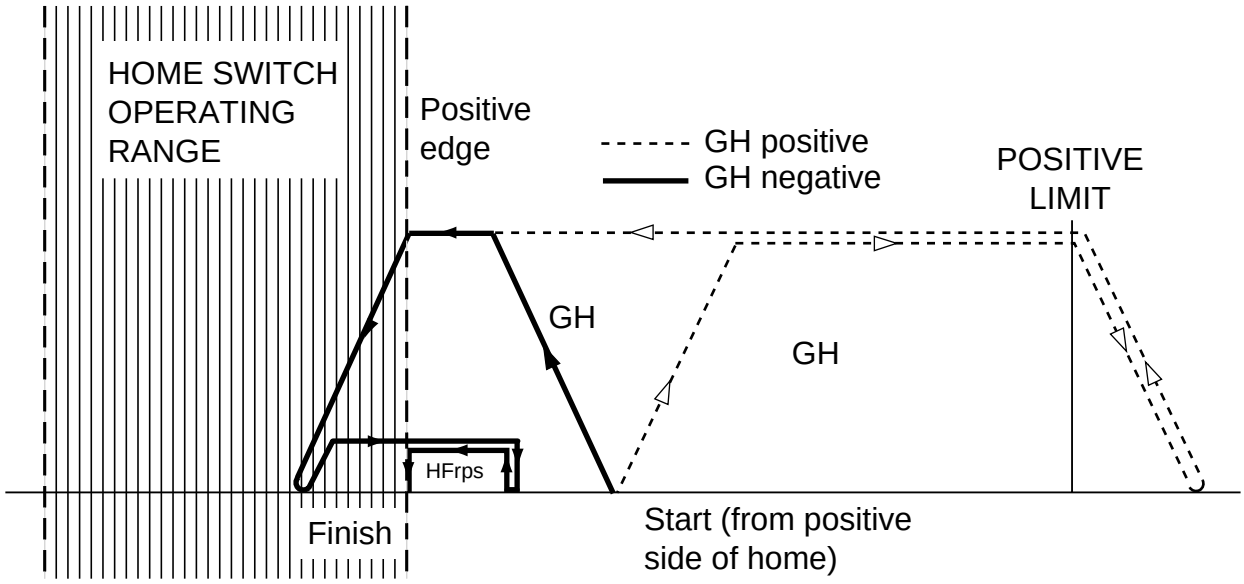


Figure 4-8. Go Home to Positive Edge

If the negative edge of the home switch is selected in the homing configuration command a similar motion path would be followed, but finishing on the other side of the home switch, as shown in Figure 4-9.

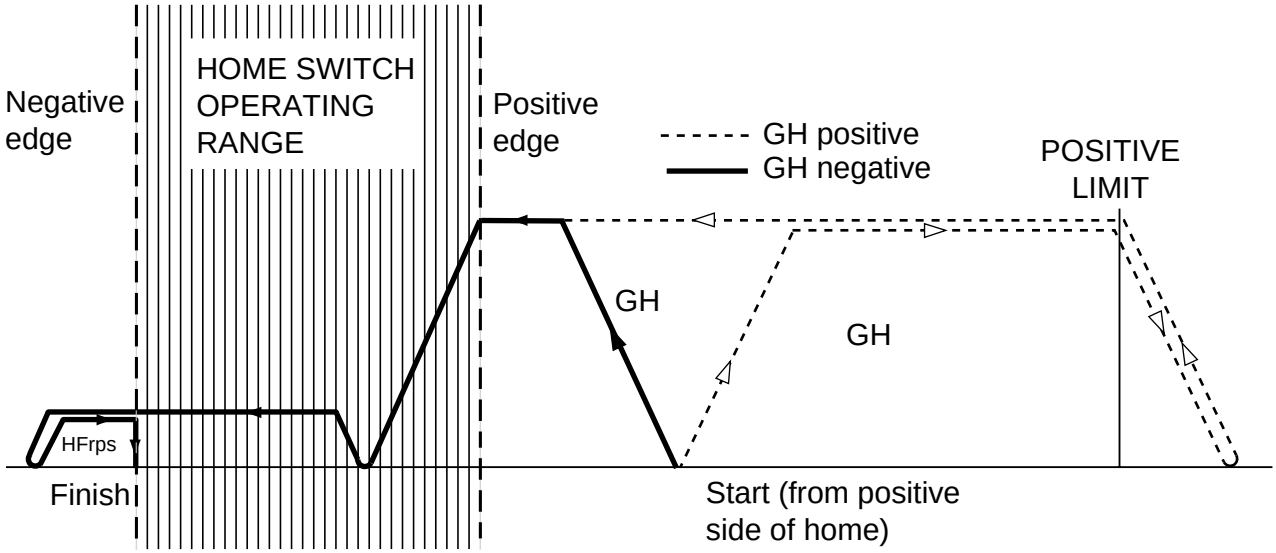


Figure 4-9. Go Home to Negative Edge

Motion starting on the negative side of the home switch will behave in a similar way, the only difference being the direction of travel. If the drive was started up already within the boundaries of the home switch and a go home command was given for a particular edge the motion would follow the path shown in Figure 4-10, depending upon which edge was requested. In this situation the home position is known so the indexer knows in which direction to travel to seek the appropriate edge. In Figure 4-10 acceleration and deceleration are set to the same value.

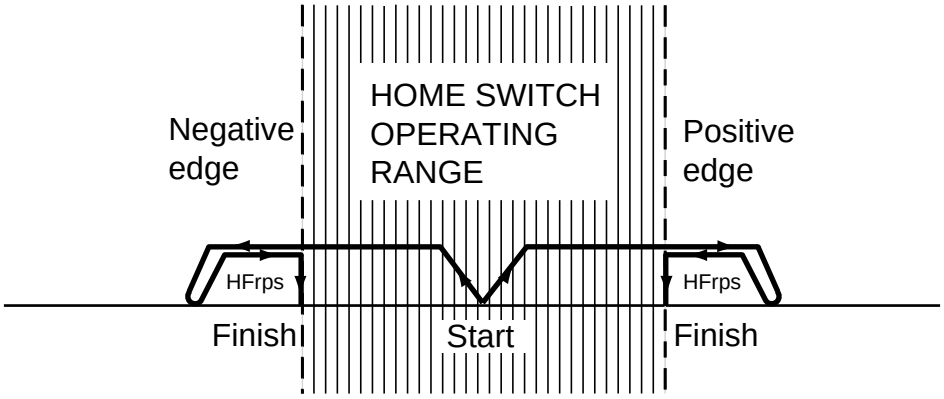


Figure 4-10. Go Home Starting from Home

Note: If the home configuration command is set to mode 0 and the home switch is already in its active range, no movement will take place.

Final Direction of Travel

Note that no matter where motion starts from, that is from positive side of the home switch, in the home switch region or from the negative side of the home switch, or in which direction it goes from its starting point (positive or negative), its final direction of travel towards a nominated home switch edge is always the same. Direction of travel towards the positive edge of the home switch is always negative and the direction towards the negative edge of the home switch is always positive. This minimises variations in the home switch operating point between separate homing moves.

Example of Homing (Datum) Routine

```
1START:                 ; start label definition
1DECLARE(MOVE3)         ; declare label
1LIMITS(0,1,0)          ; configure limits (enabled, normally closed, stop when hit).
1HOME1(+,1,-15,100,1)   ; configure the home parameters
1GOTO(MOVE3)            ; transfer to label move 3
1END                    ; end of label definition

1MOVE3:                 ; define program label move 3
1O(0)                   ; turn off o/p 1
1GH                     ; execute the go home move
1O(1)                   ; turn on o/p 1 after go home complete
1A100                   ; set acceleration to 100rps2
1V25                    ; set velocity to 25 rps
1D4000                  ; distance to 1 motor rev
1G                      ; execute move
1END                    ; end of program move 3 definition
```

Interaction Between Homing and Limits

In certain applications a limit switch may be used to define the home position, in which case one switch can be used for both a limit and the home position. This requires the consideration of two possible situations:

1. Where home and limit switches are wired separately
2. Where home and one of the limit switches are shared

In the first case, where home and limit are wired separately, the following interactions are possible:

When the load is already on a limit and it is commanded to go home, the initial direction of motion will be away from the limit and this may not be the direction set in the HOME command.

If a limit is enabled and hit whilst going home, direction of travel will be reversed (bounce off a limit) and motion will continue until the home position is reached. If a second limit is hit or the first limit is hit for the second time, the user fault ‘homing failed’ will be set and the

system will respond as if a limit has been hit in the 'normal' manner, that is, whilst not performing a homing move.

In the second case, where home and limit are wired together, the following interaction is possible:

If the load is commanded to go home in a direction away from the home switch and hits a limit, then the move will be automatically started in the opposite direction. When the load reaches the combined limit/home switch, the home function will terminate in the normal manner.

Limits

End-of-travel limits are used to restrict the movement of the load to a safe operating distance. The placement of limit switches defines the direction of motion, since positive motion is always regarded as moving towards the positive limit.

Two of the drive's user inputs (I/O 4 & 5) can become dedicated limit inputs (negative and positive) when enabled by the LIMITS command. From start-up, both limits are enabled (default setting) and can only be disabled by issuing a disable limits command. For fail-safe operation the limit switches must be normally closed, although this can be re-configured within the LIMITS command.

Limit Switch Placement

Limit switches need to be placed such that when a limit switch is hit sufficient travel is still left for the load to be decelerated to a stop before hitting the hardware limit or end stop. Hitting a limit is defined as changing the state of a limit switch when that limit is enabled and the direction of motion is appropriate, that is, you would only expect to hit the positive limit switch when travelling in the positive direction.

Hitting a Limit

When a limit is hit, an error signal is generated (*E), the user fault bit 'end of travel limit hit' is set and the status bit '+limit' or '-limit seen during last move' is set. Motion decelerates at the rate set in the LIMIT command, which needs to bring motion to a stop before any hardware limit is reached. If motion is requested whilst the load is already on the limit no motion will take place and the drive will respond as if the limit had just been hit, although no deceleration will take place.

A fault label can be made to run once a limit is hit, subject to the following conditions:

- No fault label is already running
- ARM command is armed and has the fault label enabled (**ARMX1**)
- Within the LIMIT command the mode is set to 'Stop motion when a limit is hit and abort program'
- A fault label has been defined

If no fault label is defined, or fault is not armed (within the main ARM), the program will be aborted, that is motion will be stopped at limit deceleration, the program is halted and all

associated flags are cleared. The program will also be aborted if you are already on a limit and you request motion in a direction which takes you further on to that limit.

If the LIMIT command has been set to 'stop motion when a limit is hit but continue the program' and you hit a limit or request motion in a direction which takes you further on to a limit no response will be given, apart from a possible *E (depending upon the setting of the EX variable). In this situation, program execution will continue and no fault label will be run. This allows the limit switch to be used as both a limit and home switch.

Hitting Both Limits

If both limits are hit motion will be stopped and the drive will respond as if a single limit has been hit, but no further motion will be possible until both limits have been cleared. The status will report which limit was seen first (positive or negative), but if both were hit in the same millisecond period, the positive limit will be reported as being 'seen' first.

Clearing a Limit

A limit is cleared as soon as a motion command is given that moves the load away from the limit, that is, in the opposite direction to which the limit was originally hit. Once a limit has been cleared and the limit switch has returned to its normal state (closed or open contacts) movement can be commanded in either direction.

Following and Limits

The way the drive reacts to hitting a limit while following depends upon the setting of the FOLLOW mode parameter.

In encoder following mode (E), it is possible to re-enable following on a limit and reverse off the limit. The drive will prevent motion further onto the limit while allowing motion off the limit.

In all cases, the recommended action when a limit is hit during following, is for the application to perform an indexed move to a position between the +ve and –ve end of travel limits, before re-enabling following.

Limit Switches

The drive has two limit inputs, the positive limit input and the negative limit input. When wiring the limit switches it is essential to check that a positive direction command produces motion towards the positive limit switch. If this is not the case, interchange the motor connections to A+ and A- to reverse the motor direction.

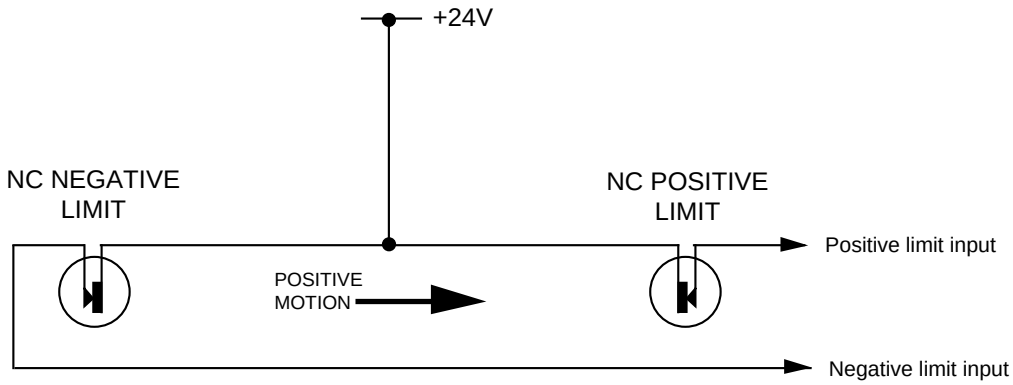


Figure 4-11. Limit and Stop Switch Configuration

If a faulty limit switch, or some other fault caused the indexer to sense both limits becoming active at the same time all motion would be stopped.

Using Closed Loop Operation

Closed loop refers to the operation of a stepper motor/drive where the position of the stepper motor shaft is measured and compared with the commanded position. This is normally arranged using an encoder attached to the motor's shaft and electrically connected to the stepper drive's encoder input, as shown in Figure 4-12.

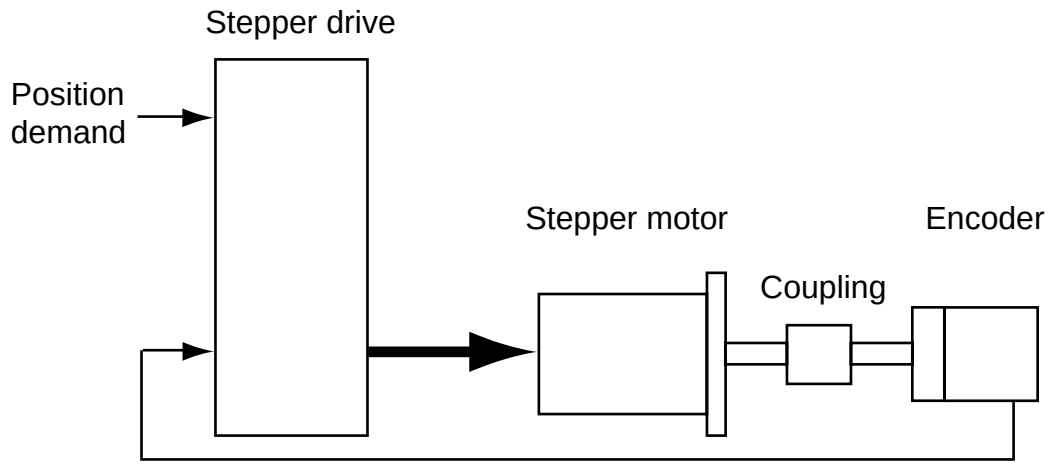


Figure 4-12. Closed Loop Operation

Closed loop operation is normally used in applications where a motor stall must be detected (stall detect) or where a known position of the motor shaft must be maintained within known limits (position maintenance).

Encoder Setup

To operate in closed-loop mode a motor- or load-mounted encoder must be connected to the primary encoder input X2 and firmly attached to the motor shaft.

When using a motor mounted encoder set motor resolution in the MOTOR command equal to the post-quadrature encoder counts per rev. See ***Scaling*** at the end of this section. When using a load mounted encoder set the system variable EM equal to the post-quadrature encoder counts per rev. See ***Scaling*** at the end of this section.

With LOADENC on (load-mounted encoder), distance is commanded in load encoder steps. With LOADENC off (motor-mounted encoder) distance is commanded in motor encoder steps.

Note: Post quadrature resolution is a hardware technique for increasing encoder resolution by a factor of 4, consequently an encoder with a 250 line count will have 1000 counts per revolution.

For a correctly connected system, entering a positive distance value should cause the motor shaft to rotate in a CW (Clock Wise) direction when viewed from the shaft end and should cause the encoder count to increase in a positive direction. The encoder can be checked by entering a positive distance value (D) and noting the direction travelled by the motor shaft. Then de-energise the motor (using the OFF command) and read the current encoder

position using the 1R(PA) command. Now, rotate the encoder shaft in the same positive direction by about half a turn. Again read the encoder position, which should be greater than the first reading, indicating that the encoder count is increasing for positive motion. If the second count is less than the first, cross over the A- and A+ signals on the encoder connector, and repeat the test until an increasing count is obtained. Encoder signal A should lead B for positive motion.

Note for a load mounted encoder, that is with LOADENC enabled, the system variable EM may be set to a negative value as an alternative to crossing over A- and A+ signals on the encoder connections.

Position Maintenance

Position Maintenance is a method of correcting occasional position errors by adding or subtracting motor steps once a move has been completed. It is not like a servo loop in which position error corrections are made throughout the entire move.

To be able to make use of Position Maintenance a drive system needs to be fitted with a load or motor mounted encoder. The drive's controller will detect the difference between the number of steps the motor was commanded to move and the number of steps actually being reported by the encoder. This resultant position error is used, at the end of a move, to further command the motor in a direction to give the correct target encoder position, as shown in Figure 4-13.

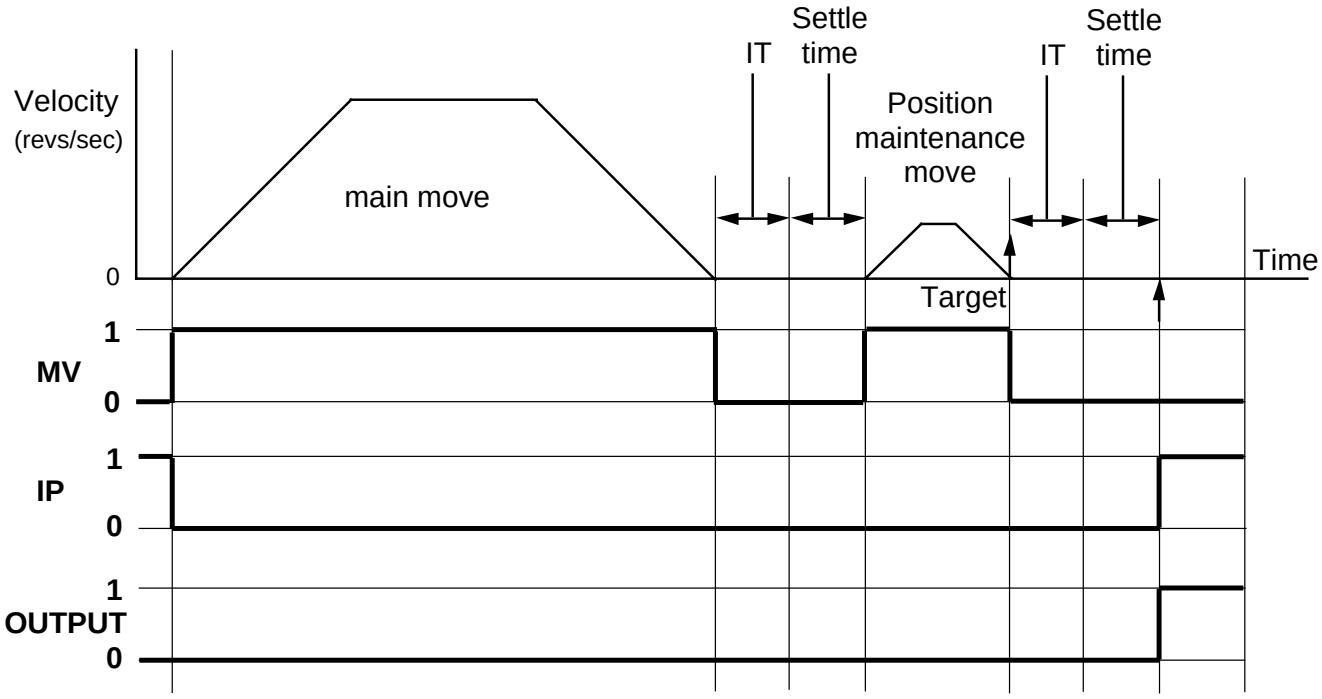


Figure 4-13. Position Maintenance Move Profile

At the end of the main move the controller waits until the in position time delay and the settle time (if programmed) have timed out, at this point the encoder count is read. A calculation is performed which compares the encoder count with the target position, if the difference between these two readings is less than the defined dead-band then the move is complete and the next user program command is executed. If the two readings differ by an amount greater than the dead-band then position maintenance is used to correct the move error.

Assuming the motor has not quite reached the target position and that position maintenance is required, the difference between the target position and the encoder count will be the number of steps yet to be moved. The indexer will automatically apply a correction move, based upon the number of steps yet to be moved, and will then, after the appropriate delays, re-read the encoder count. Once again a comparison is made between the encoder count and target position and the whole process is repeated, depending upon the result of the comparison.

Dead Band Range

With Position Maintenance enabled, if you command the motor to move one revolution, at the end of the move you would expect the encoder count to read the encoder resolution. In practice, mechanical alignment errors and lost motion within the system will usually result in a small offset existing between commanded motor steps and the encoder reading. To take account of this offset an error band is defined, known as the Dead Band Range. It has a range of 0 to 32767 encoder counts and a default value of 10. The number of counts entered must be positive, but the range will check the number of counts on both positive and negative sides of the target position. Position maintenance will have deemed to be successful if the final correction move positions the motor within the dead band range. Note: the value entered must be in load-mounted encoder steps if LOADENC is enabled, otherwise it is entered in motor-mounted encoder steps.

Output

An optional output can be used to signal when position maintenance is enabled and the motor is in position. In position is defined as not moving whilst positioned within the dead band range. See the note at the end of the example.

Settle Time

When Position Maintenance is enabled all moves will track actual position against commanded position. Position Maintenance allows the in-position signal to be held off for a settle-time, the value of which can be programmed in the command parameters.

Speed Of Correction

Once a move has been completed and the controller decides position maintenance correction is required, it will move the motor at a fixed speed of 1rps.

Example of Position Maintenance

The following code illustrates how position maintenance is implemented. The example is based upon a motor resolution of 4000 steps per rev and a 1000 line encoder giving 4000 counts per rev.

```
1DECLARE(TRIAL)
1MOTOR(X,X,4000,X,X,X,X)     ;X is set depending upon the application
1TRIAL:
1ON
1R(EI)                       ; check encoder is set to quadrature operation
1POSMAIN0(20,3,0)            ; set-up position maintenance, dead band of 20 encoder
                               steps, output 3 to be used, no programmed settle time
1D40000                      ; program distance, 10 revs
1V5                          ; set velocity to 5 rev/s
1A10                         ; set acceleration to 10 rev/s/s
1POSMAIN1                    ; enable position maintenance
1POSMAIN                     ; check status of command
1W(PA,0)                     ; set absolute position to zero
1G                           ; start the move, motor turns 10 revs
1R(PA)                       ; read position
1END
```

Following the G command the system will attempt to correct any final position error at the end of the move.

Note: When the command is armed output 3 will come on with the motor in position and stationary. When the G command is given output 3 will turn off until the motor is back in position within the dead band.

Stall Detection

Stall detection is only possible if an encoder is fitted to the motor or load. A stall is reported when the error between the commanded position and the actual position, determined by the encoder, exceeds the value set in the error window of the STALL command.

Stall Detection Set-up

A stall condition is signalled when the number of expected stall-encoder steps does not match the number of steps received. During a move the indexer constantly monitors any build-up of stall error, and once the difference exceeds a programmed error window, a stall condition is reported. Note, the stall error count is reset following an **ON**, **STALL**, **GH** or **G** command.

Set system variable EM to equal the number of stall encoder counts per rev. This allows the use of a low resolution stall-detect encoder without effecting the motor positioning resolution (as set in the motor command). However, if LOADENC is enabled the positioning resolution is now determined by EM as distance is commanded in stall encoder steps.

The error window size needs to be large enough to detect a single de-synchronisation of the motor which is the equivalent of one rotor tooth or 4 full steps (7.2 degrees). Allowing for the normal lag and lead occurring during acceleration and deceleration, of up to 3.6 degrees, an overall error window of 5 degrees should be set - 14 steps with a 250-line encoder. The error window is measured in motor steps with LOADENC and SCALE disabled, load steps with LOADENC enabled, and user steps with SCALE enabled.

Fault on Stall

When STALL is enabled (on/off parameter set to a 1), and mode is set to 1 (run fault) motion is stopped if the error between the commanded position and the actual position exceeds the error window value. If a fault label is defined for this condition a fault will be reported and can be identified by reading the status bits.

Output

Any one of the drive's outputs 1 to 3 can be turned ON when a stall condition is detected. This command option allows a stall to be signalled externally by lighting a lamp or LED.

Scaling

Using scale allows 'user-friendly' settings of distance, velocity and acceleration to be defined in user units, rather than units required by the drive. For example, using a ViXIM to control a linear table, it is possible to program distance units directly in mm, velocity in mm per second (mms^{-1}) and acceleration in mm per second/per second (mms^{-2}). This is made possible by measuring one user unit in terms of the number of positional feedback encoder steps. This measure of Position Encoder/Motor steps per (user) Unit is termed the PEU parameter. For example, a linear table with base units of 1mm and having an encoder that gives 1 count every 5µm of travel, results in a PEU of (1mm/5µm) = 200 (PEU must be => 1).

The PEU value is used with the SCALE command and once a PEU value is set this will determine the units in which acceleration, distance and velocity are measured. In this case, a base unit of 1mm was chosen, consequently acceleration is measured as 1mm s^{-2}, velocity as 1mm s^{-1} and distance in mm.

Individual scaled values of acceleration, distance and velocity can be set using:

SCLA	SCaLe Acceleration factor
SCLD	SCaLe Distance factor
SCLV	SCaLe Velocity factor

For example, to work with distance set in increments of 0.1mm set SCLD as (base unit)/(desired unit) = 1 mm/0.1 mm = 10. This will require the SCALE command to take the form:

SCALE1(1,10,1,200)

For more information see the SCALE command.

A, D and V do not have to be in the same units, any combination of units is possible, **but PEU divided by SCLD must result in an integer**. This is because the distance moved requires the following calculation:

D * (PEU/SCLD) steps, which could result in a fractional number of encoder steps that cannot be resolved by the drive.

Once defined using the **SCALE** settings command, an application can be simply programmed in user units, without needing to calculate what units the drive requires.

You can use SCALE in combination with other commands such as LOADENC, STALL or POSMAIN. The exact mix of commands together with the source of the feedback, and the type of programming steps used are presented in Table 4-9. In the command columns 0 = disabled and 1 = enabled. In the feedback source column Motor = motor-mounted encoder steps, Load = load-mounted encoder steps and X = invalid combination. In the command steps column (the steps used to program the application e.g. distance D) Motor = motor steps (1 rev = motor resolution), Load = load steps (1 rev = load resolution EM) and User = user steps with X representing an invalid combination.

SCALE	**LOADENC**	**STALL**	**POSMAIN**	**Feedback source**	**Command steps**
0	0	0	0	Motor	Motor
0	0	0	1	Motor	Motor
0	0	1	0	Motor	Motor
0	0	1	1	Motor	Motor
0	1	0	0	X	X
0	1	0	1	Load	Load
0	1	1	0	X	X
0	1	1	1	Load	Load
1	0	0	0	Motor	User
1	0	0	1	Motor	User
1	0	1	0	Motor	User
1	0	1	1	Motor	User
1	1	0	0	X	X
1	1	0	1	Load	User
1	1	1	0	X	X
1	1	1	1	Load	User

Table 4-9. Distance Units for Enabled Commands

5. Easi-V Software

Computer Requirements

To be able to run Easi-V software, necessary for the control and programming of the ViX, you will require an IBM™ compatible PC running Windows 95/98/2000/XP™, NT4 or ME.

The PC needs to be specified to run Windows™ with at least 16MB of RAM, a VGA monitor, Windows™ compatible mouse, CDROM drive. The installed program size is approximately 1.3MB. Easi-V is supplied on a CD or may be downloaded free of charge from our Website (www.parker-emd.com).

Serial Link Lead

You will need a **2-wire plus ground** lead which has the **Rx and Tx wires crossed over**. Wiring details are given in the ***Electrical Installation*** section.

Note: The information contained within this section applies to Easi-V software version 2.0 or greater. If you have an earlier version of Easi-V software please request the latest version from Parker EMD using the contact numbers given at the beginning of this user guide or download a copy from our web-site (www.parker-emd.com).

Establish Communications

Before attempting communication with the drive the supplied software needs to be installed on to the PC's internal hard disk drive. Once software installation is complete, commands can be downloaded from the PC to the drive to confirm its operation. If Easi-V has already been loaded you may skip the following Installation and Operation sections up to ***Confirming Drive Operation***.

Compatibility of EASIV

Always use the latest version of Easi-V, available from our web-site or supplied with the product.

Software Installation

Before attempting to install the EASI-V software supplied with your drive check that your PC meets the requirements previously defined under ***Computer Requirements***.
Easi-V software is supplied on a CDROM or can be downloaded from the Parker website and installs in the usual manner common to Window™ applications.

Installation Procedure

This procedure takes you quickly through the steps necessary to install Easi-V on your PC. The entire installation process takes less than 10 minutes. Before starting the installation, terminate all applications currently running.

A step by step installation of Easi-V software follows:

1. Place the Easi-V CD in your PC's CDROM drive.

2. Once loaded the CD should auto-start. If this does not happen, open the CD's folder and double-click the **EMD.exe** icon.

3. Follow the on-screen instructions to load Easi-V.

4. The screen will display the Easi-V program banner and will prepare an installation setup program.

5. The banner screen is automatically replaced by a Welcome dialogue box advising you of the need to exit any programs currently running. To abandon setup in order to exit other programs, select CANCEL. This in turn displays an Exit Setup dialogue box giving you the options of Exit Setup, which returns you to Windows™ or Resume which takes you back to the Welcome box.

6. Selecting NEXT> displays a 'Choose Destination Location' dialogue/selection box that provides the option of installing Easi-TOOLS in the directory of your choice. The default directory is **c:\program files\parker\easi-v** in the UK, but the exact path name is country dependent, other buttons are described within the dialogue box, see Figure 5-1.

Figure 5-1. Choosing Where to Install Easi-V

6. Once you have selected a destination for Easi-V or have decided to use the default directory, select NEXT to begin file transfer.

7. Once Easi-V has been loaded, the screen will display a message dialogue box, stating 'Setup is complete. You may run the installed program by double-clicking on the program icon.' When you click the OK button, the window shown in Figure 5-2 will appear. Note: Easi-V may also be run from the Start menu.

Figure 5-2. Easi-V Application Window

Uninstalling Easi-V

To uninstall Easi-V software, use Windows™ uninstall software facilities available within the Control Panel. All components are removed.

Software Operation

Once installed, Easi-V can be started from the start menu or by double clicking its application icon. At startup Easi-V displays the product selection screen shown in Figure 5-3. Select the micro-stepping drive, either with or without CANopen.

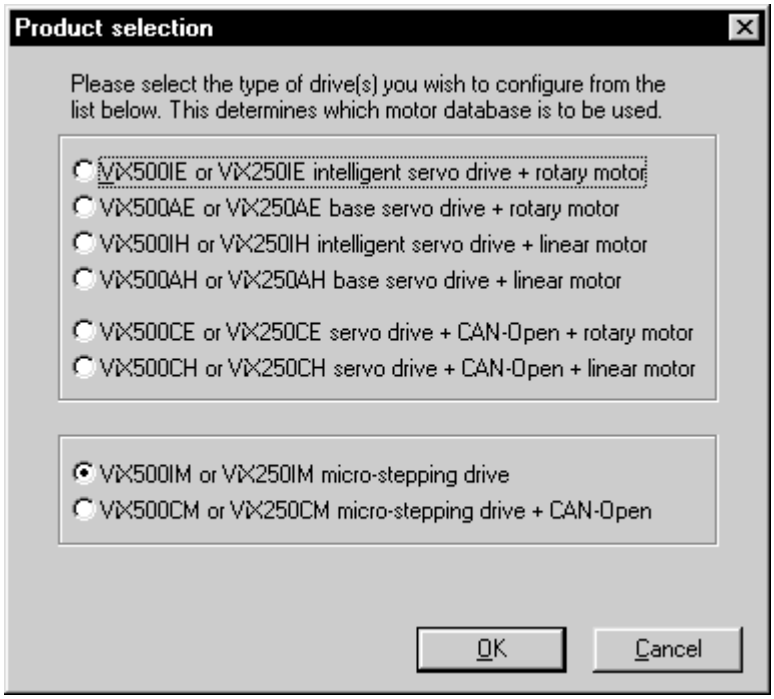


Figure 5-3. Easi-V Product Selection

Selecting Product from the Utilities menu will also display the product selection screen.

Selecting OK will display the main application window, entitled
'Parker Hannifin EMD – Easi-V', and seven pull-down menus become available:

File, Edit, Search, Terminal, Utilities, Windows, Help

The majority of options available within each menu are familiar to Window™ users and will not be fully described here, but options available within Terminal and Utilities are specific to drive control and will be fully described.

Menu Overview

File

Filing Operations

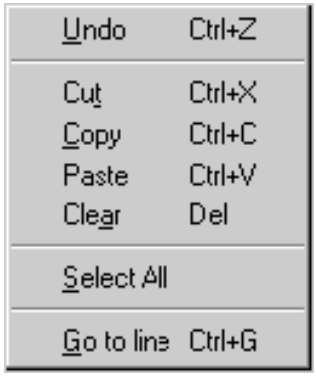
Creates a new editor file, or .prg program file
Opens an existing editor file or program
Save an editor file
Save an editor file specifing the file name

Print the editor file or contents of terminal buffer

Close current active window

Exit Easi-V

Edit



Editing Operations

Undo a previous edit (1 level of undo only)

Remove highlighted text to clipboard
Copy highlighted text to clipboard
Paste contents of clipboard to current cursor location
Delete highlighted text

Highlight all text in active editor file window

Go to a particular line within a file*

Search

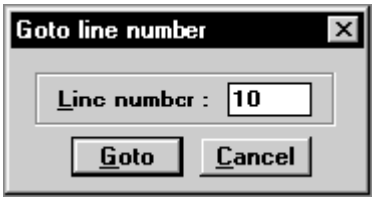
Editor search & replace operations

Find specified text (top down from cursor position)

Repeat search again

Find and replace text (top down from cursor position)

* Selecting 'Go to line' from the edit menu will generate the following dialogue box, which allows the required line number to be entered. This is useful for locating errors when loading a program.



Terminal

Terminal on-line operations

Configure the serial communications

Open/close the terminal (after testing the connection)

Configure test buttons

Edit buffer Create an editor file window or terminal buffer

Log to file Open/close file logging terminal buffer actions

Utilities

Specific tools

Displays Product Selection screen

Help screens guide you through drive setup

Similar to above, but uses tabbed text boxes

Display status bits or messages

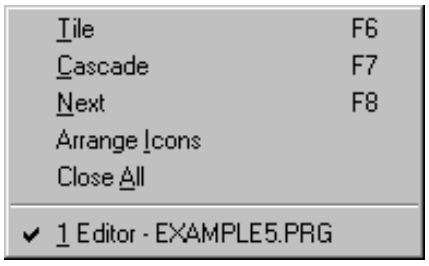
Download program to drive F4 Download program

Upload program

Displays LED diagnostics sheet, colour & flash rate

This provides a variety of tools documented below.

Windows



Window controls

Share program desktop space between open windows

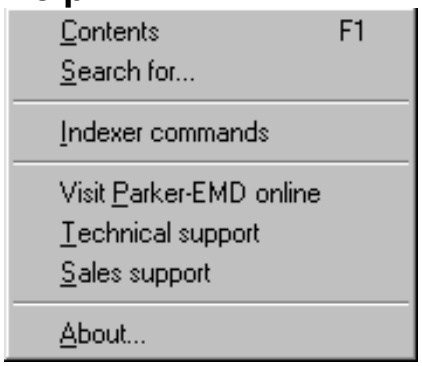
Cascade all open windows on program desktop

Select/activate the next window

Arrange all minimised windows on program desktop

Close all active windows on program desktop

Help



Program help facilities

Open help file at the main contents (start)

Prompt for topic string and search help file

Open help for individual EASI-V commands

Visit Parker web-site

E-mail Parker technical support

E-mail Parker sales support

EASI-V version number and copyright

Utilities Menu Options

Selecting Options displays a single screen with three tabs:

- General
- Drive settings
- Country

Use **General** to select the following set-up options:

- Prompt to save terminal emulator contents on closure
- Test communications before upload/download of programs
- Display 3 rows of buttons in terminal window

Use **Drive settings** allows the following selection:

- Automatically upload parameters on address change

Use **Country** to select your preferred Parker contact:

- Germany
- Italy
- United Kingdom
- USA

Terminal Menu Selections

Terminal menu selections control the setup and configuration of communication between a PC and drive.

Communicating with a Drive

The default settings of a new drive from power-up are RS232 communications with an address setting of #1. Wire the RS232 communication lead as described in the Electrical Installation section.

WARNING

To avoid causing damage to a PC serial port the drive must be earthed before making any serial connections.

Connect the RS232 lead between the drive's X3 socket and the controlling PC's RS232 connector.

Configure the serial communications

From the Terminal menu choose Settings to display the following Communications port configuration dialogue box.

The default settings used are:

Port	COM1
BAUD rate	9600
Options	Auto wrap

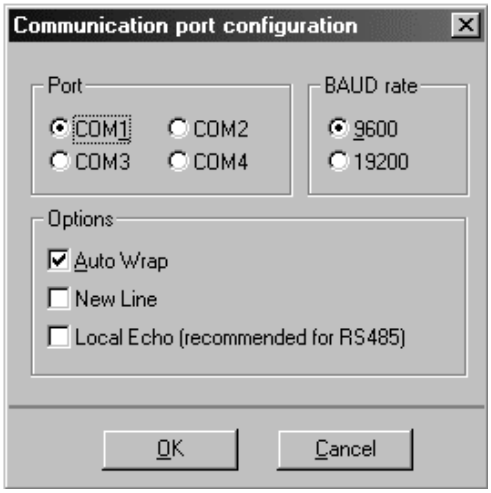


Figure 5-4. Comms Port Configuration

Select the required configuration and, click OK. Then, again from the Terminal menu select Connect to start communications. Every time Connect is issued the communications link is tested to establish it is working correctly and the message box 'Testing communications integrity' is flashed on the screen, followed by 'Now on-line to controller' if the Connect is successfully made. If the link fails, refer to the Troubleshooting Section. Note: The baud rate selected must agree with the drive's hardware selected value.

You are now ready to start creating and editing program (.prg) files to control the operation of a connected drive. A number of example files are included within the Easi-V installation package to give you a start with drive programming.

Configure Terminal Buttons

The Terminal menu has a 'Configure terminal buttons F2' command which generates the window shown in Figure 5-5.

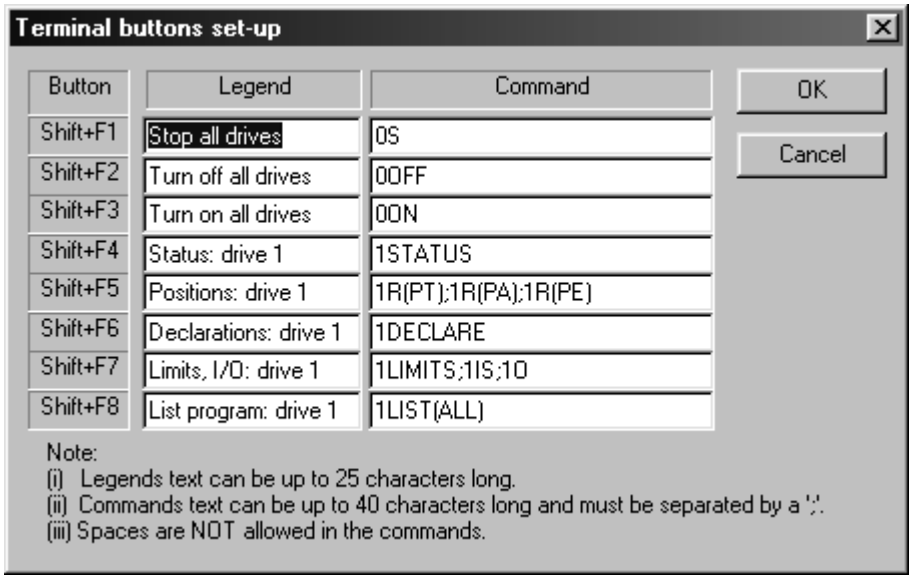


Figure 5-5. Configure Terminal Buttons Window

Defines the function of the buttons at the base of the Terminal window and assigns a keyboard shortcut to each button, depending upon where it appears in the list order. This facility enables a group of commonly used commands to be sent to a drive(s) by clicking a single button in the Terminal window or pressing a Shift/Function key combination from the keyboard.

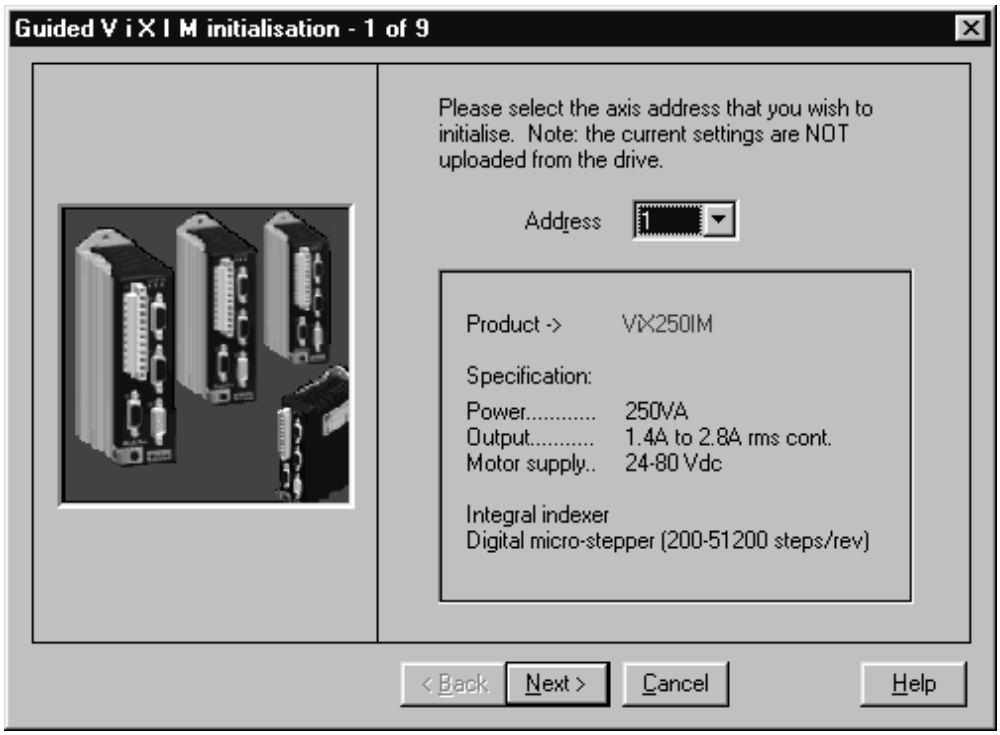
Utilities Menu Selections

Utilities menu selections control the way drives are setup and configured for use with a particular motor type. The menu offers two levels of setup, depending upon the skill and experience of the operator.

- Guided drive setup (guides you through setup for a particular motor type – quick and simple)
- Drive setup (allows text entry of motor parameters – for experienced users)

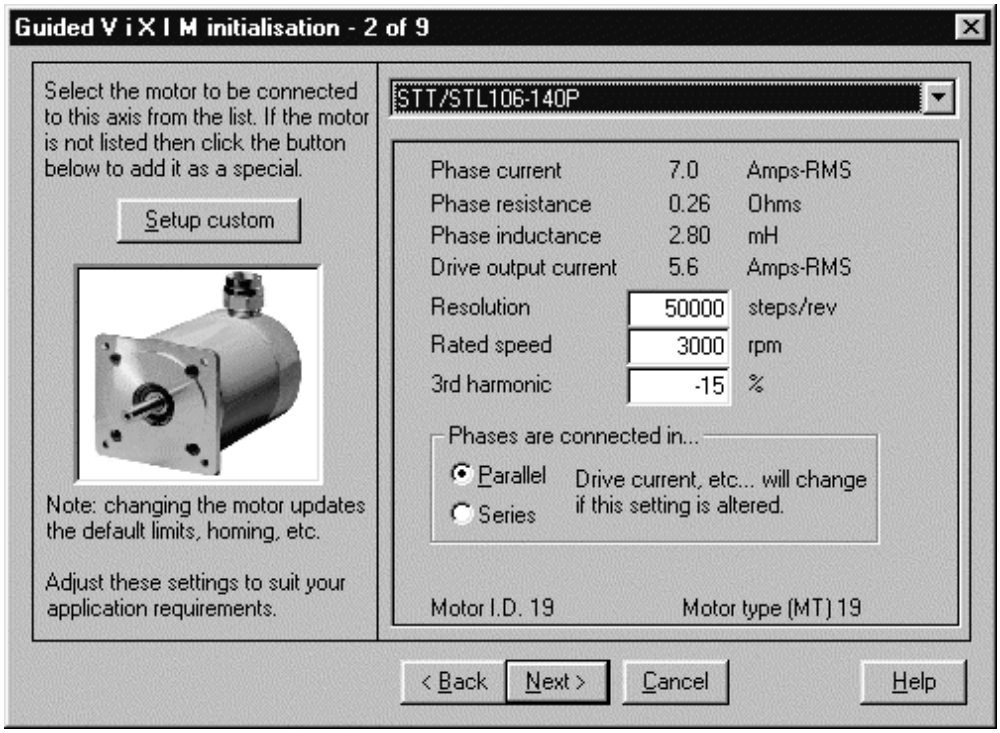
From the Utilities menu select 'Guided drive setup'.

Select the axis address of the drive to be initialised.



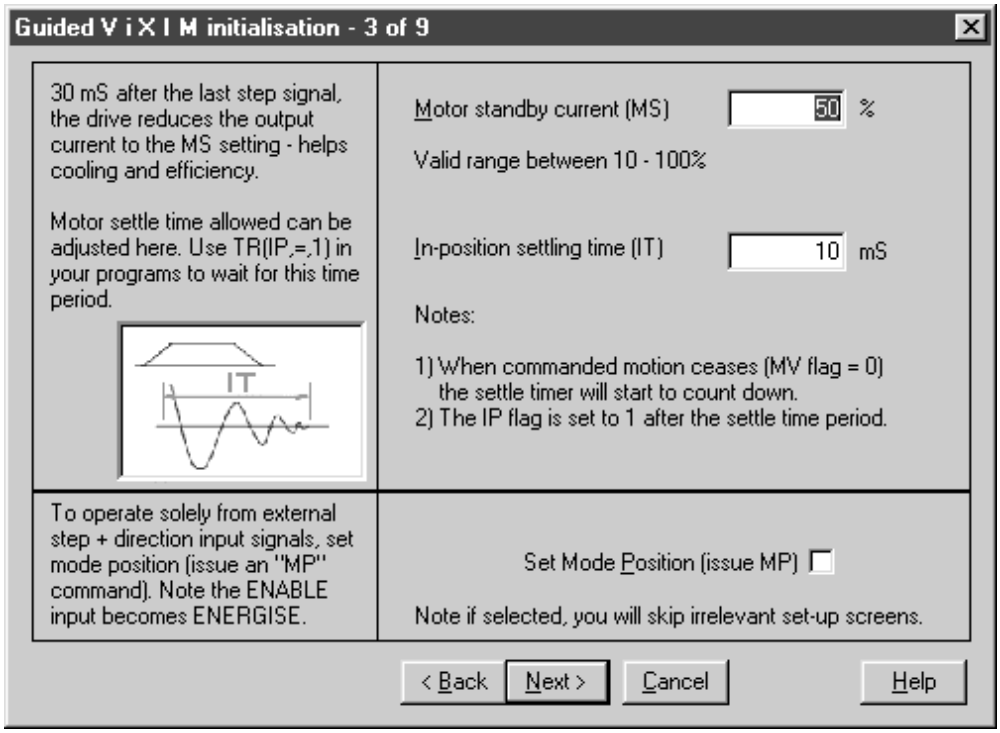
Press 'Next' to select the required motor type.

From the drop-down menu select your motor type or perform a custom set-up.



Press ‘Next’ to select Motor Standby current and In-position settling Time values.

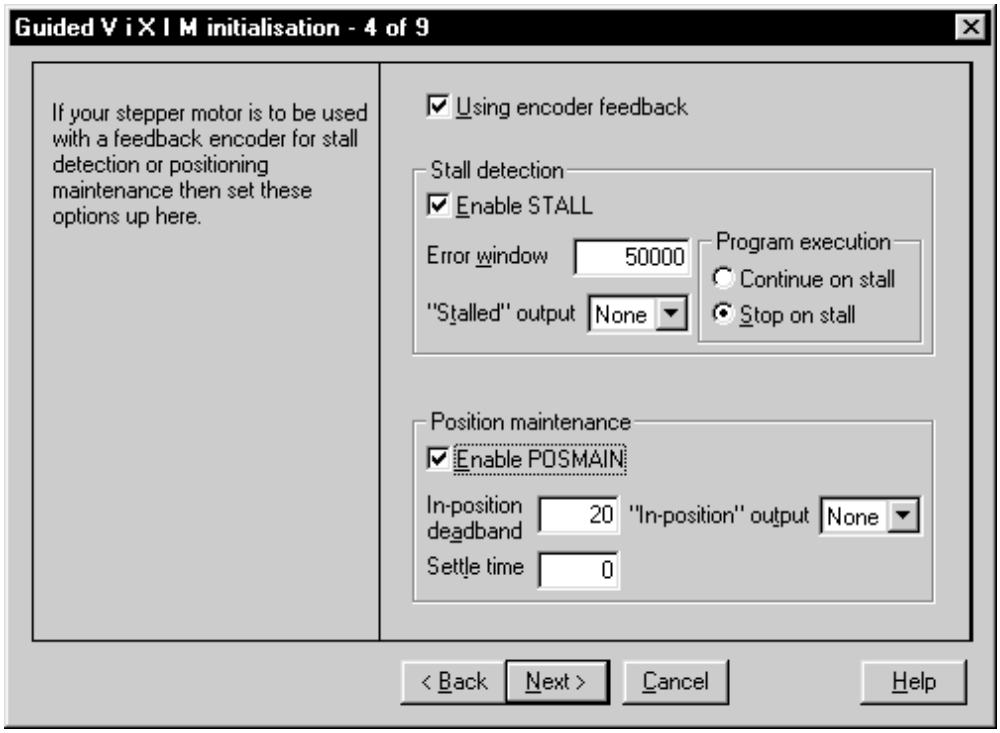
This screen allows the selection of the motor standby current (MS) and the In-position settling time (IT).



Press ‘Next’ to allow selection of stall detection and/or position maintenance.

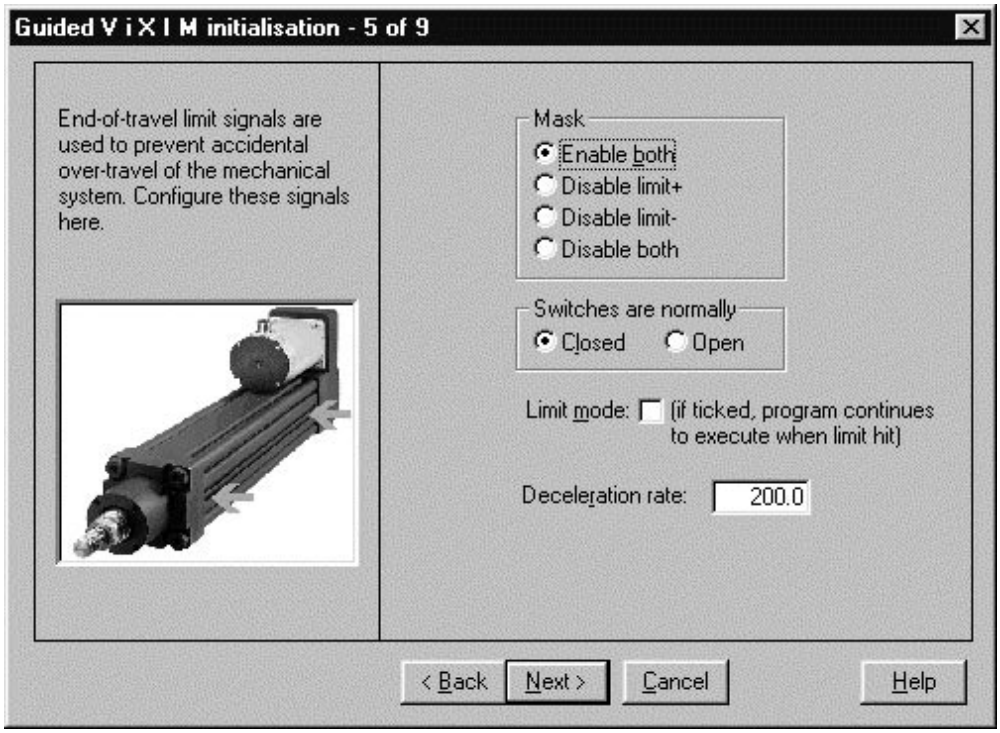
Selecting Set Mode Position jumps forward to guided screen 8.

Encoder feedback must be selected to allow stall detection or position maintenance.



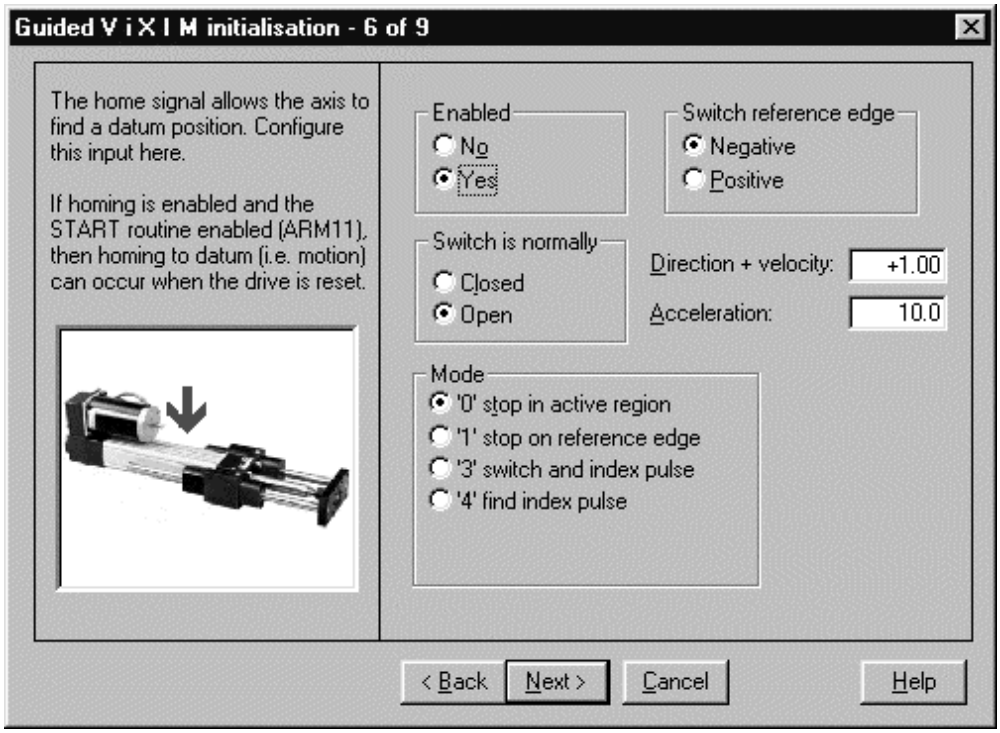
Press 'Next' to select the required limits.

This screen allows the selection of the + & - limits and the type of limit switch used. You may also control the limit mode and the deceleration value used to bring movement to a stop.



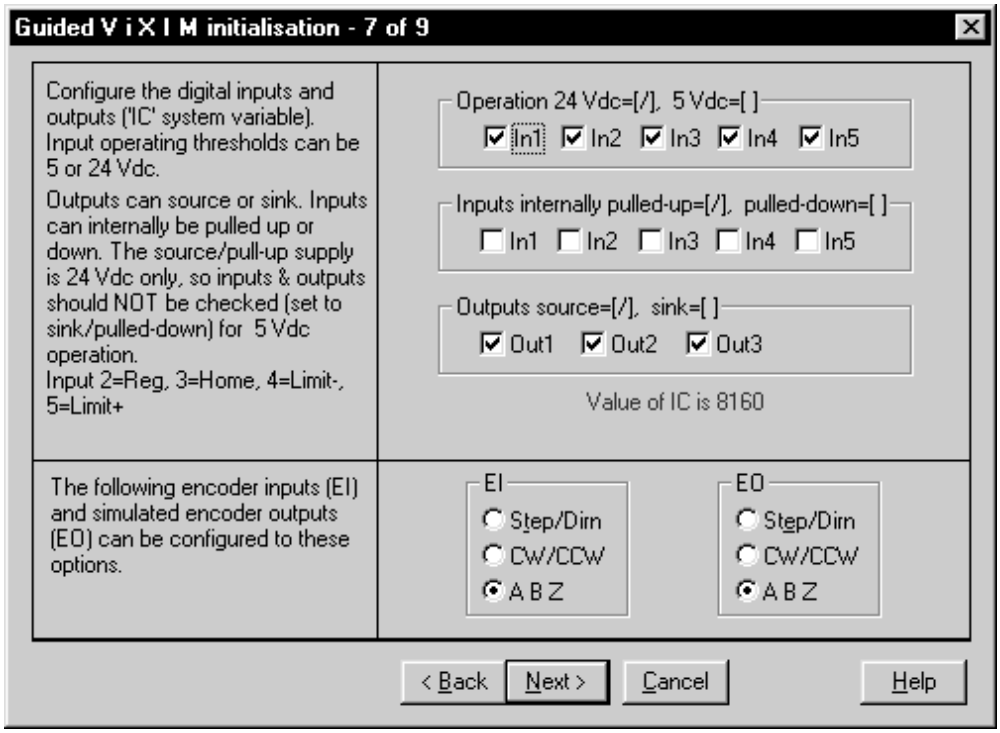
Press 'Next' to configure the home switch setting.

This screen allows the configuration of the home switch.



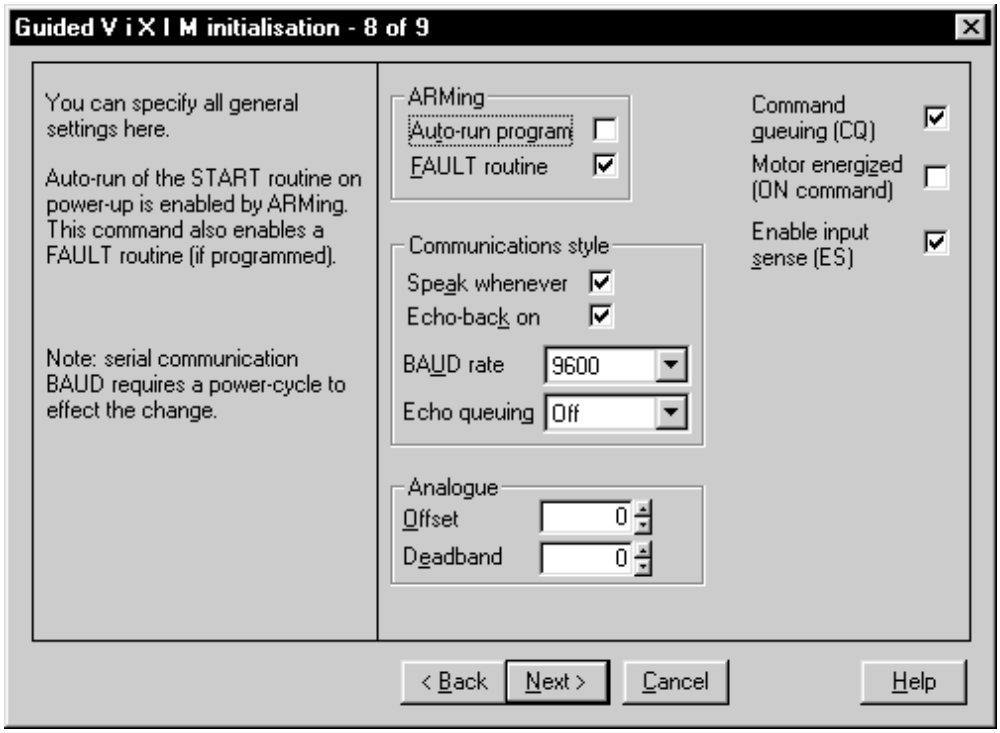
Press ‘Next’ to configure the drive’s user inputs and outputs.

Configure the drive’s user inputs and outputs and setup the following encoder inputs or the simulated encoder outputs.



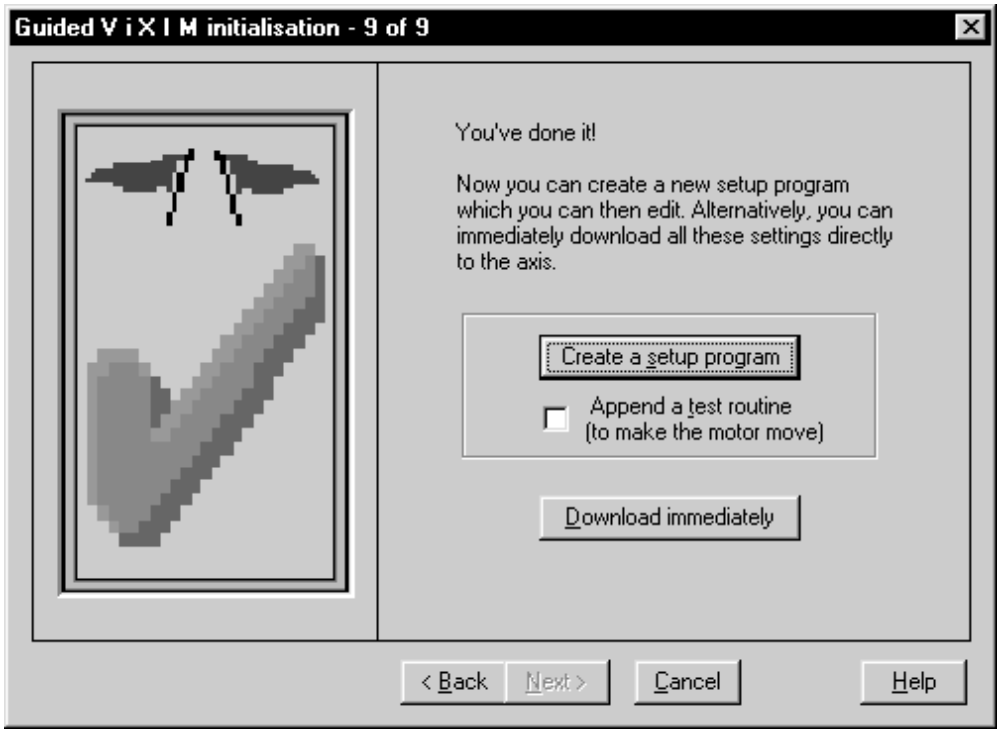
Press ‘Next’ to configure the drive’s general settings.

This screen allows the setup of all the drive's controls not available on other screens.



Press 'Next' to complete the setup process and to generate a setup program.

This screen signals the end of the guided setup process.



Checking the 'Append a test routine' box will include a simple routine that turns the motor shaft to verify drive operation. To alter any configuration set-up step backwards using the 'Back' button.

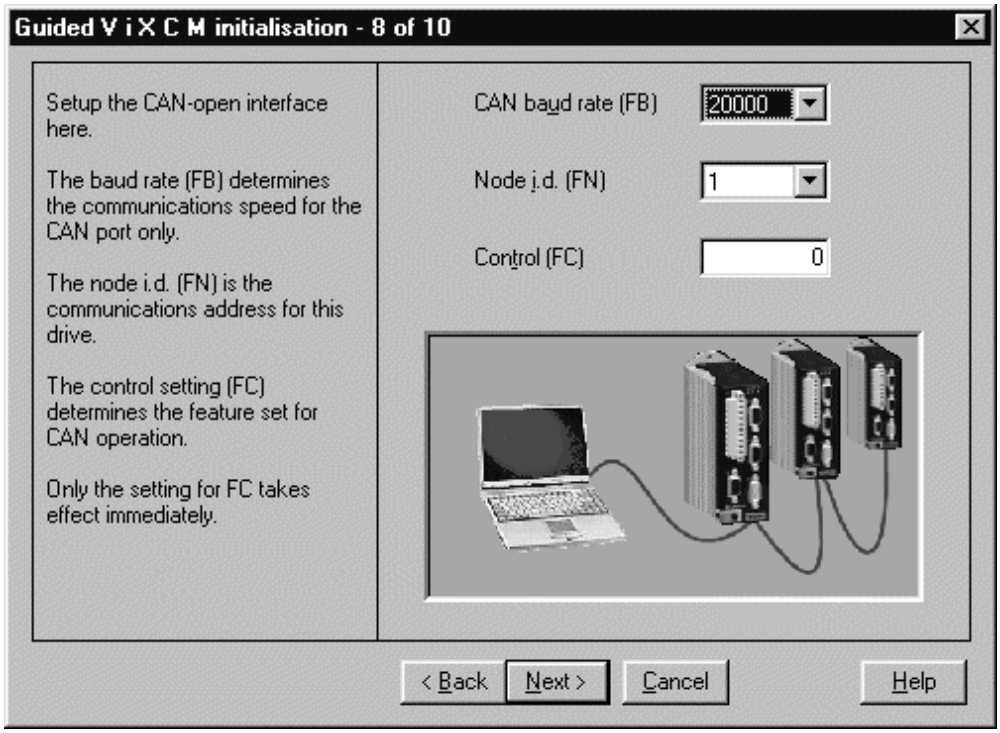
NOTE

ANY CHANGES TO THE MOTOR TYPE NUMBER MUST BE FOLLOWED BY A SAVE (SV) AND RESET (Z) OR CYCLING POWER TO THE DRIVE.

TIP
Create a setup program first, rather than download immediately, because if changes are required it's easier to edit a saved program.

CANopen Drives

If you are setting up a CANopen drive an extra screen is included (8 of 10) that allows adjustment of the baud rate, node ID and control setting.



Drive settings / Setup

This facility gives easy access to setting system variables in a more direct manner than Guided Drive Setup. Figure 5-6 shows a sample screen.

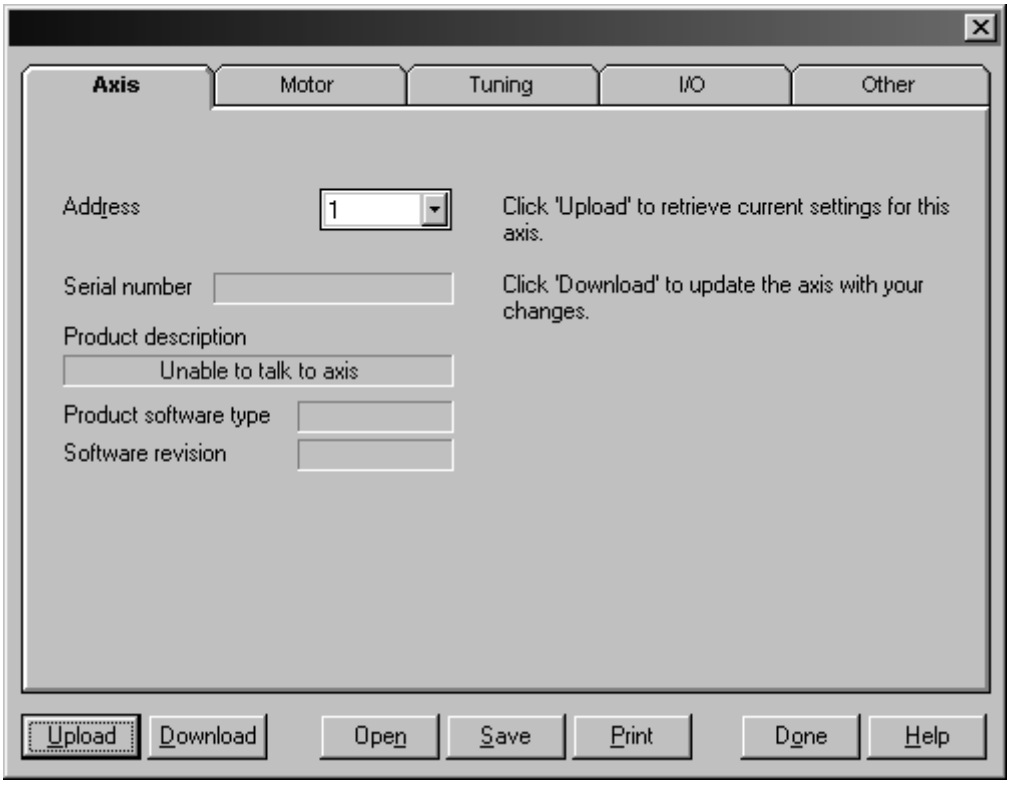


Figure 5-6. Axis Setup Tab

The buttons displayed along the base of this screen can be used as follows:

Upload retrieves the current settings for the selected axis

Download updates the current axis with your latest changes

Open opens a stored .cfg file

Save save the file to disk as a .cfg file

Print prints the value of all the settings of all the tabs

Done closes the set-up window

Help accesses the help file

Status

The Utilities menu axis Status provides a convenient method of examining the double word status bits. The tool gives access to the status of User Faults, Status bits and Drive Faults using a series of tabs, as shown in Figure 5-7.

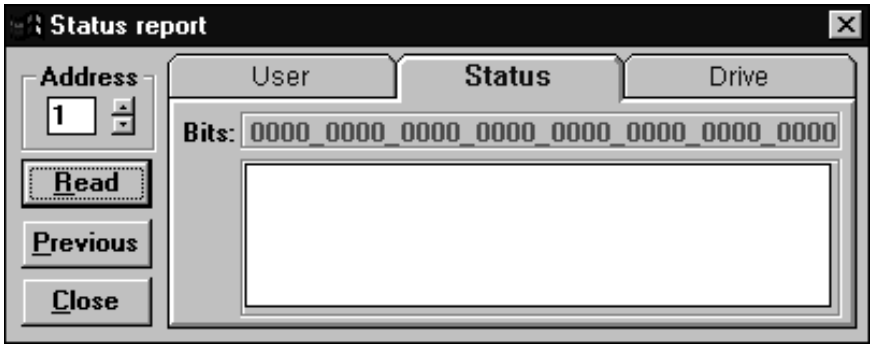


Figure 5-7. Status Reporting

The Status Report can be permanently displayed during program development or testing to monitor the operation of the drive. The double word status bits are decoded and displayed as text messages within the Status Report window, as shown in Figure 5-8. This eliminates manual decoding errors and gives an immediate update of the drive's status.

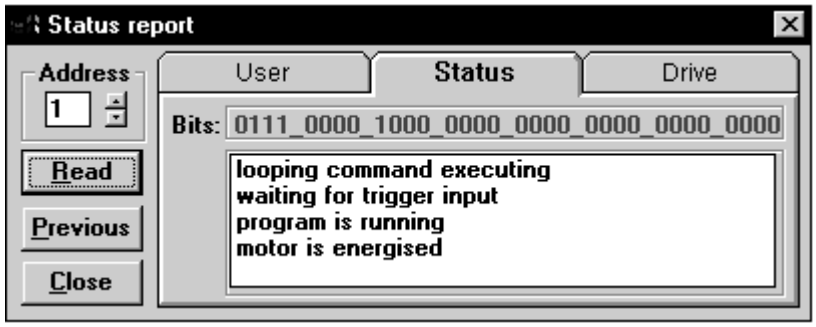


Figure 5-8. Reporting Status of Drive

Read

The Read button is used to update all of the Status Reports and is a useful aid when debugging an application.

Previous

Selecting Previous allows the previous status to be re-displayed - useful for comparing the results of programming actions. The Previous reading is only stored to a depth of one, that is, you cannot trace the history of status bits by continually selecting the button.

Close

Selecting Close will exit the Status Report window.

Downloading and Uploading Programs

A drive program that exists within an active edit window can be downloaded to the drive by selecting Download from the Utilities menu. Following the usual communication checks, the program will download to the drive's internal memory. Function key F4 provides a shortcut download.

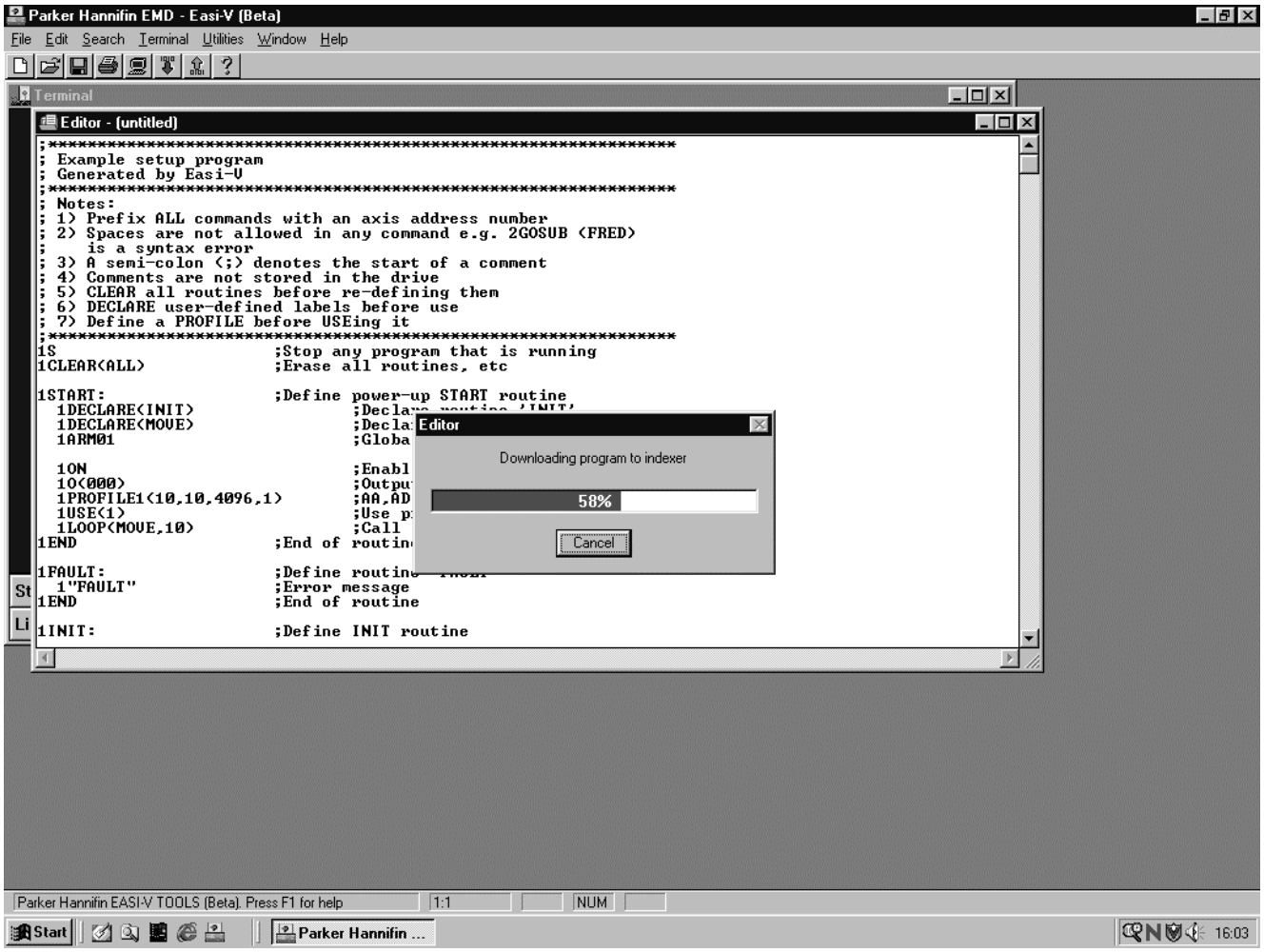


Figure 5-9. Download a program to the drive

A program may also be uploaded from a drive, a useful facility if a drive needs to be swapped between axes. To upload a program select Upload from the Utilities menu. An upload dialogue box will be displayed, allowing you to specify the name and address of the program to be uploaded. A shortcut upload is provided by function key F5.

Figure 5-10. Upload Dialogue Box

If you receive an error message during program upload refer to the Troubleshooting Section.

Help

EASI-TOOLS has extensive on-line help facilities, which allows you to search for help on a particular topic either within the main contents or by entering a topic string. All the commands listed within this user guide are available on-line by selecting Controller Commands from the Help menu.

Confirming Drive Operation

With the drive and motor correctly wired and the serial connection made to a PC running Easi-V software, the operation of a drive may be confirmed by creating and downloading the following code:

Before running this code return the drive to its factory settings and save those settings – see returning a drive to its factory settings in ***Maintenance & Troubleshooting*** section.

```
*Define motor using MOTOR command*
1START:          ; define start label code
1ON              ; enable the drive
1LIMITS(3,0,0)   ; disable limits
1D4000           ; set distance to 4000 steps
1V1              ; set velocity to 1rev/s
1A10             ; set acceleration to 10rev/s2
1G               ; start motion
1END             ; end definition of start block
1GOTO(START)     ; execute start code block
```

WARNING

Clamp the motor in a secure position before testing the drive.

Upon execution of this code, the motor should perform 4000 steps and stop. The successful operation of this code confirms the drive is working correctly. If this does not happen, refer to the ***Troubleshooting*** Section.

6. Command Reference

Command Description

Each command has a simple 1 to 7 character name usually an abbreviation of its full descriptive title. Listed commands are in alphabetic order with any non-alphabetic symbols appearing last.

Each individual description will include a one-line header giving the abbreviated name followed by its full name. The following lines give the command syntax, units of measurement, range of values, any default value and a reference to other related commands. Where commands contain a list of parameters, a simple layout displays only the syntax of the command.

Every command requires an address. Where several drives need to respond to a common set of global commands, prefix each command with the address 0. To prevent spurious feedback any report or read command using address 0 will be ignored. Note a drive will ignore a command missing an address prefix.

Where commands (such as IF, R, TR, and W) include a system variable it is treated as a command parameter. System variables store internal drive values and settings. Each variable is capable of being read and tested, and some may be written to, but they are all dedicated for a particular use by the system and cannot be used for storing user data within a program.

Command Syntax

Generally, a command's syntax consists of an address 'a' followed by the command name. Parenthesis containing the commands' parameters or simply the range parameter 'n' follows this. Within the parenthesis form of command, a comma separates each parameter and italics indicate any optional parameters.

Commands not requiring any parameter string have the syntax shown in Figure 6-1.

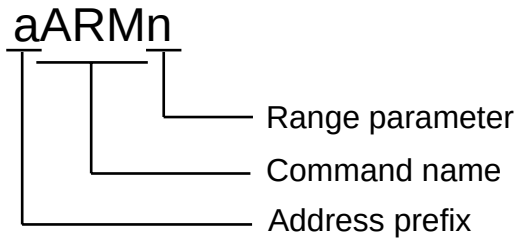


Figure 6-1. Simple Command Syntax

Commands, which include a parameter string, can be simple one parameter commands such as GOSUB or CLEAR where the single parameter is a label, or multi-parameter commands containing a string of parameter values. Figure 6-2 shows both styles of parameter commands.

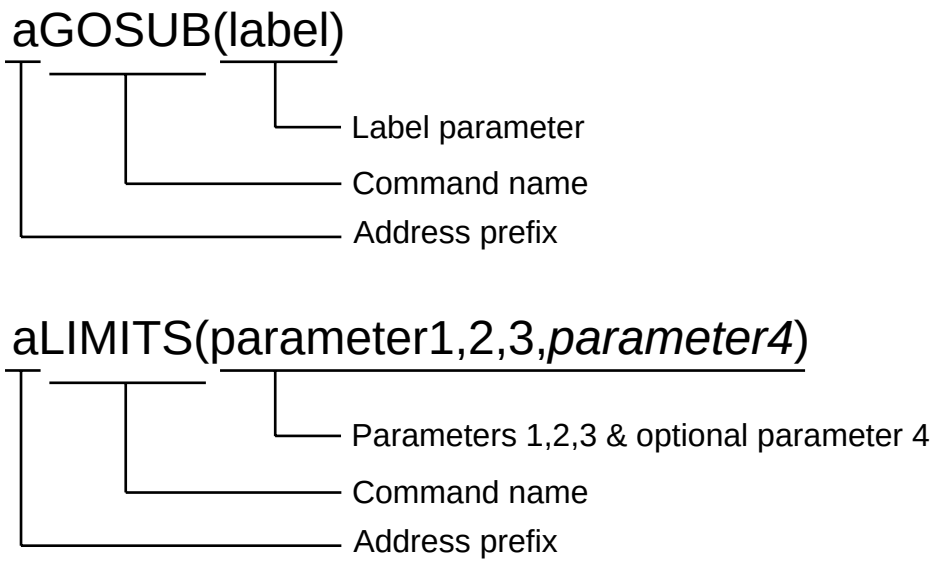


Figure 6-2. Parameter Commands

Attention

[1] Terminate all commands with a carriage return. A space is not valid.

[2] A command must not contain any space characters.

[3] All commands are device specific, that is, they always need an address.

[4] For reasons of clarity, program examples appear as if being downloaded via Easi-V, that is they contain comments and line feeds etc.

Command Properties

Each command has a particular set of properties that govern the way the command can be used.

Commands can have the following properties:

Immediate only
Immediate or buffered
Can be used in labelled block
Can't be used in labelled block
Saved by SV
Not saved by SV

Immediate Only

Immediate only commands are:

C, K, S, R(RB), R(UF), R(DF) and R(ST)

The controller acts upon these commands as soon as they are received.

Immediate or Buffered

Immediate or buffered commands are immediate unless command execution is being delayed or command queuing is enabled whilst moving, in which case the command is buffered. When command execution is being delayed, that is awaiting the results of a trigger command or waiting for a pause or time delay to finish, interrogation commands can be sent and get executed immediately. Consequently, the report of such commands as 1IS, 1R(ST), 1R(UF), 1A etc. is correct. However, if a buffered type of command is sent (such as G or 1A10) the buffered command just sent and any other interrogation commands get buffered and will not report back until the trigger, time delay or pause is finished. In this situation 'Immediate only' commands can be executed.

Example

```
1R(EI)                          ;read encoder input
*2
1IS                             ;read input status
*01111                          ;
1TR(IN,=,X0X01)                 ;input trigger false
1A50                            ;send buffered command
1R(EI)                          ;report commands are now delayed until the trigger
1IS                             ;command is complete
1R(RB)                          ;an exception R(RB) is always immediate
*1                              ;busy
.
.
(Trigger becomes true, 1A50 actioned)
*2                              ;report commands completed, EO=2
*00001                          ;new input status reported
```

Can be used in labelled block

Means it is possible to include the command within a labelled program block. Running the labelled block code will execute the command. Note, if power is removed from the controller without sending a save (SV) command the program and its labelled block will be lost.

Can't be used in labelled block

Means it is not possible to include the command within a labelled program block.

Saved by SV

A command that has the property of being 'saved by SV' means data associated with that command is capable of being stored in non-volatile memory. The saved value will become the default value on power-up or following a Z command.

Not Saved by SV

If the command does not change data, such as GO or STOP, the commands' properties are listed as 'not saved by SV'.

Automatic Checking of Valid Commands and Parameters

All commands and parameters are checked for syntax and parameter limits at data entry. Certain commands will only report an error on execution, for example, commands defined within a label. If a fault is detected, that command or parameter will be ignored during execution of the program.

For example:

1USE(2) where profile 2 has not been defined

Upon entry, this will cause a *E, *cannot use an undefined profile* type error report.

However, using the same command within a label:

```
1DECLARE(TST)     ;create a label
1TST:             ;begin the label code
1USE(2)           ;attempt to use undefined profile – no error reported
1END              ;terminate label
1GOTO(TST)        :run label code TST
*E                ;error reported at run-time
```

In this case, the same error report message is given. Note: in both cases the program will ignore the **USE(2)** command, but will continue execution using values taken from **PROFILE(0)**.

A Acceleration/Deceleration

Syntax	Units	Range of 'n'	Default	See also
aAn	see SCALE	0.01 to 99999.99	10	AA AD SCALE

Description This command will set both the acceleration and deceleration rates of the motor to the same value. Values set for the **AA** and **AD** commands are over-written, if previously set.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

To set the acceleration and deceleration rates of axis 1 to 120 rps^2, type **1A120**
To determine the acceleration of axis 1, type **1A**
The response is **120.0 120.0*
Overrange value **1A100000**
Will be reported as ..... **E* (meaning error)

Note For all error reports refer to ***Section 4 - Reporting System Information During Code Development***.

AA Acceleration

Syntax	Units	Range of 'n'	Default	See also
aAAn	see SCALE	0.01 to 99999.99	10	A AD SCALE

Description The **AA** command will set or report the programmed linear acceleration rate of the motor. The acceleration value assigned to the **AA** command is over-written, if previously set.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

To set the acceleration rate of axis 1 to 120 rps^2, type ... **1AA120**
To determine the acceleration of axis 1, type **1AA**
The response is **120.0*
Overrange value **1AA100000**
Will be reported as ..... **E* (meaning error)

AD Deceleration

Syntax	Units	Range of 'n'	Default	See also
aADn	See SCALE	0.01 to 99999.99	10	A AA SCALE

Description The **AD** command will set or report the programmed linear deceleration rate of the motor. The deceleration value assigned to the **AD** command is over-written, if previously set.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

To set the deceleration rate of axis 4 to 320 rps^2, type.... **4AD320**
To report the current deceleration rate of axis 4, type..... **4AD**
The response is..... **320*
Overrange value..... **1AD100000**
Will be reported as ..... **E* (meaning error)

ARM Enable label triggered code

Syntax	Units	Range of 'n & m'	Default	See also
aARMnm	**-**	**0 or 1**	**01**	**START label** **FAULT label**

The **ARM** command allows you to enable (arm) or disable the START label. It also enables/disables the FAULT label.

n=1 : start label is enabled
n=0 : start label is disabled (default condition)

The second parameter 'm' is a fault switch that enables/disables the fault label from being run. See ***Fault Label*** in ***Control of ViX Drives***.

m=1 : fault label is enabled (default condition)
m=0 : fault label is disabled

At power on, when saved and armed, the controller will execute the code following the START: label (if defined).

The fault label parameter (fault switch), when enabled, will call the FAULT label when any one of the following conditions occur:

1. When driving further onto a limit, whilst the limit mode is set as stop on limit and the fault switch is enabled.*

2. When hitting a limit during a move, whilst the limit mode is set as stop on limit and the fault switch is enabled.*

3. Having a hardware drive fault with the fault switch enabled.

*Note: If, within the LIMIT command, the mode is set to '1' (stop when a limit is hit but continue the program) motion will be stopped at the programmed limit deceleration. No FAULT label will be called and the program will continue in a normal manner.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example The code following the START label will be run at power up:

```
1START:                          ;start label
1T0.5                            ;delay
1ON
1LIMITS(3,0,0)                   ;disable limits
1PROFILE1(100,100,4000,25)       ;define profile 1
1USE(1)                          ;use profile 1
1G                               ;execute profile 1
1END

1ARM1                            ;arm the start label
1SV                              ;save the code
```

Note If you save the controller with **ARM0**, then the start-up sequence will fail to run, and the controller will wait for serial commands.

Using EASI-V software, certain commands become armed when their **on/off** parameter is set to '**on**'.

Requesting an ARM status will report the state of the START and FAULT labels, for example:

```
aARM
*START 0
*FAULT 1
```

C Continue

Syntax	Units	Range of 'n'	Default	See also
aC	-	-	-	PS

Description The **C** (continue) command causes a user command to resume execution following a pause command.

Properties Immediate only, can't be used in labelled block, not saved by SV

Example

```
1PS          ;pause commands
1A100        ;acceleration 100rps2
1V20         ;velocity 20rps
1G           ;go
1"*TEST"     ;add TEST comment
1C           ;continue
*TEST        ;message TEST is displayed
```

CLEAR Clear user code

Syntax	Units	Range of 'n'	Default	See also
aCLEAR(label)	-	-	-	DECLARE

Description The **CLEAR** command deletes user program instructions from the **label** specified until the END statement associated with that label. If a subroutine has been cleared, but its associated GOSUB command still exists, at run time the code will halt, motion will stop and *E will be reported.

Specifying the ALL keyword as the label will delete all user programs within the drive addressed.

Properties Immediate or buffered, can't be used in labelled block, saved by SV

Example

```
0CLEAR(ALL)      ; Clear memory of anything defined so far (all drives)
5CLEAR(START)    ; delete the power on code, but nothing else, in
                 ; axis 5
```

Note You can only clear declarations by using CLEAR(ALL).

D Distance

Syntax	Units	Range of 'n'	Default	See also
aDn	See SCALE	-2,147,483,647 to 2,147,483,647	-	M SCALE

Description The **D** command will set or report the programmed move distance. The value programmed is only used for preset moves. In MC (Move Continuous), the direction is observed.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example To set the move distance of axis 2 to 15000 steps type .. **2D15000**
To report the current programmed move distance of axis 2, type **2D**
The controller responds with **15000*

If a value entered is out of range *E will be reported and the current value will not be altered.

Distance reports the current direction as influenced by the H command in MI (Mode Incremental) only. For example:

```
1MI      ;mode incremental
1D4000   ;set distance to 4000 steps
1D       ;report distance
*4000    ;value reported
1H-      ;change direction
1D       ;report distance
*-4000   ;value reported
```

Declare Declare

Syntax	Units	Range of 'n'	Default	See also
aDeclare(label)	-	-	-	CLEAR

Description All labels, apart from START, REG, NOREG & FAULT need to be declared at the beginning of the program using a **DECLARE** command. Labels consist of up to 5 upper case alphanumeric characters terminated with a colon (:), but a label must begin with an alpha character. Choose a name that is relevant to the operation being performed, or a system label name. To terminate a code block use 'END' (no colon).
You can use up to 20 labels, although four of these have already been allocated to START, REG, NOREG and FAULT, leaving sixteen for general use.

Only declare labels in the command line or inside the START label. If you wish to upload your program all declarations must be made within the START label.

If a label has been declared, but not defined, a run time error will be signalled when it is called by a **GOTO**, **GOSUB** or **LOOP** command.

When a label has been declared and defined, clearing it will only get rid of the definition, the declaration will remain. Declarations can only be cleared using a **CLEAR(ALL)**.

Typing **aDECLARE** by itself will list the percentage of memory used by each label type.

Properties Immediate or buffered, can be used in labelled block (but only within the START label), saved by SV

Example

1DECLARE(CUT2) ;declare label CUT2
1DECLARE
**START 0.8%*
**REG 0.0%*
**NOREG 0.0%*
**FAULT 0.0%*
**CUT2 0.0%*

E Enable/Disable Communications

Syntax	Units	Range of 'n'	Default	See also
aEn	-	0 or 1	1	-

Description The **E** command disables or re-enables serial communications. E1 turns command communications ON, whilst E0 turns it OFF. In E0 mode the only command seen by the controller is E1.

Properties Immediate or buffered, can't be used in labelled block, E1 saved by SV

Example An example of when to use this command is if global commands are being sent but one axis needs to ignore them whilst all others action them.

To enable axis 6 to accept commands over the RS232 serial link, type **6E1**

To disables communications using the RS232 serial link, type **6E0**

EXIT Exit from loop

Syntax	Units	Range of 'n'	Default	See also
aEXIT	-	-	-	LOOP

Description The **EXIT** command will terminate a loop function at the end of a label.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
4START:                    ; signifies this is the power on sequence
4DECLARE(GRIP)
4PROFILE2(100,100,4000,25)  ; define profile 2
4LOOP(GRIP,0)              ; repeat the grip/ungrip code forever
4O(XX1)                    ; set output 3
4T5                        ; If we exit the loop then we pulse output
4O(XX0)                    ; 3 to indicate gripper tension out of tolerance
4END                       ;

4GRIP:
4USE(2)                    ; Use motion parameters from profile 2
4G                         ; do the move
4O(1XX)                    ; trigger the measurement
4H                         ; change to ungrip/grip
4T1                        ; delay for 1 sec
4IF(IN,=,X0X)              ; gripper tension out of tolerance
4EXIT                      ; abort loop
4IF(IN,=,X1X)              ; else continue the loop
4G
4END
```

Note In the example shown above, once the EXIT command is encountered the cycle in progress will be completed, then the code will be returned to the line immediately following the LOOP command. That is, output 3 is pulsed for 5 seconds to indicate the gripper tension is out of tolerance.

FOLLOW Configure following

Syntax **aFOLLOWon/off(source,mode,scale)**

Description The Configure Following command is used to setup following, allowing an axis to copy movement from another drive.

The **source** parameter specifies the device to follow. The only option is:

E – encoder input

The **mode** parameter determines the type of following move that takes place:

mode 1 The position of the motor follows the following source

All other mode values are reserved at present.

The **scale** parameter specifies the scaling applied to the following source. This can range from 0.1 to 500% or –0.1 to –500% in steps of 0.1%. Negative values reverse the sense of the following input.

With the following source set to “E” (encoder) the command following position indicator (system variable PF) increments when pulses are received by the following encoder input port according to the relationship:

PF = number of pulses received since following enabled * **scale**

PF accumulates in value at a rate proportional to the incoming pulses. In this mode, turning following OFF, resets PF to zero.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Note **Only mode 1 is implemented at present.**

Do not use mode absolute (MA) while following.
Do not perform a go home (GH) while following.
If you hit a limit whilst following, movement is immediately stopped and following disabled*.
Do not use POSMAIN while following.
Refer to ***X4 Connector*** in ***Electrical Installation*** section.

*Driving through a limit, disables following. In this situation, perform an indexed move back to a known reference point and investigate why the limit was passed before re-enabling following.

FRATE Feed Rate Override

Syntax	Units	Range of 'n'	Default	See also
aFRATEn	-	0 or 1	0	M

Description Feed Rate Override, the **FRATE** command, is used together with the analogue input to scale the peak velocity of the drive (V). The purpose of the command is to allow the speed of the process being performed by the drive to be controlled by a single external analogue signal. The signal is sampled at the start of each move and is used to scale the target velocity.

The resolution of the control is 1% (0.1V) of the analogue input voltage range (10V = 100%). If the analogue voltage drops below a level equivalent to 1% of the target peak velocity (the velocity being requested by the analogue input voltage), the value used will be taken as 1% of the user set peak velocity (velocity set by the V command). This is applied to values as low as 0.01rps, the minimum velocity allowed.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
1ON                 ; energise the drive
1V30                ; set the user peak velocity as 30rps
1FRATE1             ; enable feed rate override
.
At this stage, apply an analogue voltage (say 1.85V) to the differential analogue input.
.
1G
```

The actual velocity used is given by:

$$\frac{\text{(rounded down (analogue input X 10) X user set peak velocity)}}{100}$$

$$\frac{(1.85 \times 10) \times 30}{100} = \frac{(1.8 \times 10) \times 30}{100} = 5.40\text{rps}$$

Note: If the calculated result is less than 1% the value used is held at 1%.

Note In the example shown above, because of the rounding down, an input voltage of 1.89V would also give a speed of 5.40rps.

A unipolar input signal is required (0 to +10V), any voltage with a negative polarity will be regarded as 1% of full scale value.

G Go

Syntax	Units	Range of 'n'	Default	See also
aG	-	-	-	PS S K M

Description Issuing a **G** command starts motion using the parameters specified by the V, A or AA/AD, and D commands or via the PROFILE and USE commands. The mode of motion must have been previously set as this determines which parameters are used and which are ignored. For example, mode continuous will ignore the distance parameter.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1PROFILE3(150,200,1500,25)      ;define profile 3
1USE(3)                         ;use profile 3
1G                              ;perform profile 3
```

Note If no motion occurs after G is issued, the cause can be determined by using the R(UF) command. Refer to the section on system parameters for more information.

GH Go Home

Syntax	Units	Range of 'n'	Default	See also
aGH	-	-	-	HOME S K

Description The go home command instructs the controller to search for the home position as defined by the home input switch. For this command to function correctly, the home function must define the homing parameters.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1START:                 ;START label definition
1HOME1(+,1,-15,100,0)   ;define home
1GH                     ;go to datum position
1END
```

Note If no motion occurs after **GH** is issued, the cause can be determined by using the R(UF) command to report faults. Refer to the section on system parameters for more information.
Whilst going home, registration (if armed) will be disarmed and on successful completion of the **GH** routine the registration armed state will be restored.

System variable HF sets the home final velocity.

GOSUB GO to SUBroutine

Syntax	Units	Range of 'n'	Default	See also
aGOSUB(label)	-	-	-	GOTO

Description The **GOSUB** command continues user program execution from the **label** specified and once the END statement is reached (in the called code), program execution returns to the calling routine.
GOSUBs can be nested to a maximum of 16 times, although the number of nestings will be decreased if used in combination with a **LOOP** command.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1DECLARE(MOVE1)
1DECLARE(MOVE2)
1START:                          ; code run after power on
1PROFILE1(360,360,400000,20)     ; define some move profiles
1PROFILE2(360,360,400000,45)
1GOSUB(MOVE1)                    ; go do move 1 and come back
1GOSUB(MOVE2)                    ; go do move 2 and come back
1O(1XX)                          ; set output 1
```

```
1END

1MOVE1:
1USE(1)                    ; use the move profile 1
1G
1O(XX1)                    ; turn output 3 on
1T0.1                      ; wait for 100mS
1O(XX0)                    ; turn output 3 off
1END

1MOVE2:
1USE(2)                    ; use the move profile 2
1G
1T1                        ; pause for settle time
1TR(IP,=,1)
1END
```

Note If you exceed the number of nesting levels the program will halt and return a **E*. R(UF) will return a 'Program nesting overflow' message.

If a **GOTO** command is used, the number of nesting levels is set to zero.

GOTO GO TO routine

Syntax	Units	Range of 'n'	Default	See also
aGOTO(label)	-	-	-	GOSUB

Description The **GOTO** command continues user program execution from the **label** specified.
Program execution does not return to the original place in the program (use GOSUB if command execution is required to return).

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
2DECLARE(MOVE1)                          ; declare move 1
2DECLARE(MOVE2)                          ; declare move 2
2START:                                  ; code run after power on
2PROFILE1(360,360,400000,20)             ; define some move profiles
2PROFILE2(360,360,400000,45)
2GOTO(MOVE1)                             ; perform move 1
2END

2MOVE1:
2USE(1)                                  ; use the move profile 1
2G
2GOTO(MOVE2)
2END

2MOVE2:
2USE(2)                                  ; use the move profile 2
2G
2GOTO(MOVE1)
2END
```

Note If a sequence that is being looped, executes a GOTO command, the loop is terminated.
The example shown above will give endless motion, only a FAULT condition or an immediate KILL or STOP command via comms. would stop this program.

H Change direction

Syntax	Units	Range of 'n'	Default	See also
aHn	-	+ - or blank	+	D, LOOP

Description The **H** command changes the direction of motion. Specifying H+ sets the direction to clockwise, H- counter clockwise, and H alone reverses the current direction.

This command has no effect in Mode Absolute.

In Mode Continuous, the use of H+ and H- are recommended for setting direction.

If H is entered whilst the motor is moving, the direction will not change until the motor comes to a stop and another G command is given.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example 1

Refer to LOOP

```
3MI                          ; mode incremental
3A20                         ; accel and decel to 20
3V15                         ; max speed of 15 rps
3D-8000                      ; 2 revs ccw
3G                           ; move
3H                           ; go cw next time
3G                           ; 2 revs cw

3START:
3PROFILE1(360,360,40000,20)  ; define profile 1
3MI                          ; mode incremental
3GOTO(MAIN)
3END

3MAIN:
3USE(1)                      ; use profile 1 parameters
3G                           ; do the move (CW)
3H                           ; change direction (CCW)
3G                           ; go back
3END                         ; end of user program
```

Note ***CAUTION***

The USE command or the D command will re-define the move direction each time it is executed. To set up a loop to go CW (clockwise) then CCW (counter clockwise), make sure the D or USE command is outside of the LOOP otherwise the direction will be the same each time around the loop.

HOME Configure Homing

Syntax **aHOMEon/off(reference_edge,home_type,direction_&_velocity, acceleration/deceleration,mode)**

Description The Configure Homing command is used to setup homing prior to the use of the **GH** (Go Home) command. See also system variable HF.

Use **on/off** to arm and disarm homing.

The **reference edge** parameter is used to select the required edge of the home switch (+ for edge nearest the CW (positive movement) limit or - for edge nearest the CWW (negative movement) limit), see Homing section.

The **home_type** parameter is used to select the type of switch to be used for homing, the choice is:

Home switch normally open	0 (default)
Home switch normally closed	1

Direction and velocity determines the direction in which home is initially searched for and the velocity at which homing is performed.

Acceleration/deceleration sets the acceleration and deceleration rates used.

The **mode** parameter determines what happens when the specified edge of the home switch is encountered:

- mode 0 The controller positions the motor in the active window of the switch (default setting).
- mode 1 The motor is positioned to the required edge of the switch + or -.
- mode 2 Reserved.
- mode 3 If an encoder with a Z channel is used then the controller will seek the Z position after detecting the specified home switch edge.
- mode 4 If an encoder with a Z channel is used then the controller will seek the Z position without the need for a home switch.

For linear encoder applications there is normally only one index (Z) position. Mode 4 should be used to save the use of a home switch.

For rotary applications where the maximum distance required is one revolution the index mark may be used as a unique home position. For all other applications mode 3 should be used as the index position will not be unique.

In mode 0, when the home position is reached, the absolute position of the controller is set to 0. The incremental position reports the distance moved to reach the home position. (system variable PI).

Typing **aHOME** on its own will return the current parameter values for the nominated axis. For example:

The command **3HOME** will return:

** AR1 E- TP1 V+10.00 A10.0 M0*

Meaning the command is armed, reference edge is negative, home type is 1, velocity is 10 rps positive, acceleration is 10 rev/s^2 and mode is 0.

Once configure homing has been setup, it can be applied (turned ON), or armed using the simplified form of command:

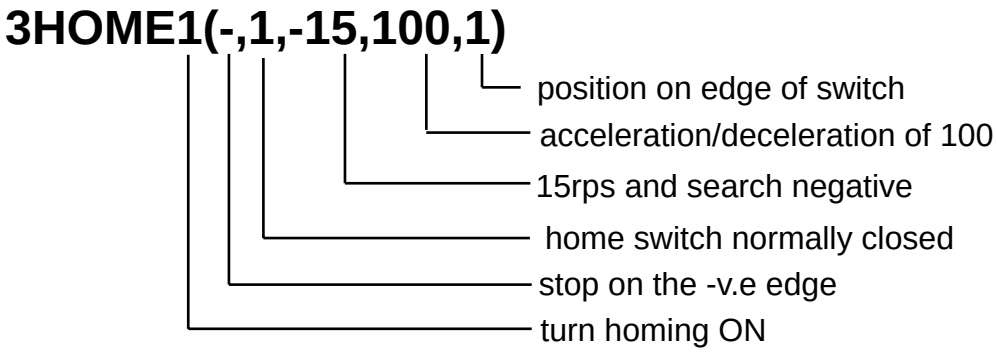
aHOME1

Or turned OFF using:

aHOME0

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example On axis 3 search for home in the negative direction at a velocity of 15 rps and acceleration/deceleration of 100 rps^2. The motor is to stop on the negative edge of the home switch and then seek zero phase.



The go home final velocity of 0.1 rps is used to complete the last part of the move.

Note Also see GH command.

IF Test condition

Syntax **aIF(system_variable,relation,value)**

Description The **IF** command compares the specified **system variable** with the specified **value** using the specified **relation**. If the condition is met, the next line of code is executed otherwise it is skipped.

Refer to the table of system variables that can be used for conditional control.

Valid relations for the comparison are:

=	Equals
<>	Does not equal
>	Greater than
<	Less than

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
2IF(PA,>,450)      ; if absolute controller position > 450 steps on axis 2
2O(1XX)            ; set output 1
2IF(PA,>,500)      ; if absolute controller position > 500 steps
2O(X1X)            ; set output 2
```

Using inputs

```
2IF(IN,<>,1X00X)   ; if input does not match the pattern
2O(XX1)            ; set output 3
```

Note If you wish to use the **IF** command during motion, command queuing (system variable CQ) must be set for continuous execution (CQ=0).

IS Input Status

Syntax	Units	Range of 'n'	Default	See also
aIS	-	-	-	O

Description The **IS** command reports the status of the configurable user inputs when configured as pull-down (non-inverting) inputs.

When used to report the state of any input switch, regardless of how the switch is wired, that is as a pull-up or as a pull-down:

0 represents an open contact switch input
1 represents a closed contact switch input

The response is five (0 or 1) digits corresponding to the following input bits:

X3 Pin Number	Bit	Function
10	1	User input 1/Stop
9	2	User input 2/Reg
8	3	User input 3/Home
7	4	User input 4/LIM-
6	5	User input 5/LIM+

0 represents a low condition on the input (configured as a pull-down input)
1 represents a high condition on the input (configured as a pull-down input)

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example To check the input status of axis 1, type **1IS**
The response is..... **01100*

User inputs 2 and home input 3 are high. All other inputs are low.

K Kill

Syntax	Units	Range of 'n'	Default	See also
aK	-	-	-	S, PS, KILL

Description Issuing a **KILL** will command motion to stop at limit deceleration rate. The command will then zero the position error to remove torque from a stalled load.

Carefully consider the use of this command in applications where a load with a large inertia may be required to stop quickly. By commanding K the motor could stall and lose torque. For this reason, a load with large inertia should be stopped mechanically to avoid overshoot of limit switches. Power dumping may be required to protect the drive from over voltage trips.

For a controlled stop use the **S** (stop) command.

The KILL command cannot be used in a label, its use is primarily for emergency situations.

K will also terminate any program execution and will disable a FOLLOW command.

Properties Immediate only, can't be used in labelled block, not saved by SV

Example

```
1G      ;set drive in motion
1K      ;stop everything
```

Note The **K** command does require a device address or 0K to kill all axes. It will stop a time delay (**T** command) and will abort a program.

CAUTION

KILL uses the LIMITS command optional deceleration rate LD, if this is not set a default value of 200 rps^2 is used.
BEWARE THAT A LOW SETTING OF LD COULD RESULT IN THE MOTOR TAKING LONGER TO STOP AFTER A 'K' THAN AFTER AN 'S'.

This command does not replace the requirement for an additional hardware device to cut power to the motor in an emergency.

LIMITS Configure limit inputs

Syntax **aLIMITS(mask,type,mode,*LD*)**

Description The **LIMITS** command allows the user to define whether the LIM+/- inputs are used as limit inputs or user inputs via the **LIM_MASK**.

The **mask** field takes the following values:

0	Enable limits (default setting)
1	Disable limit +
2	Disable limit -
3	Disable limit + & -

The +ve limit switch is the switch that is reached when the motor reaches the end of travel for a move with +ve distance/velocity (CW).

The -ve limit switch is the switch that is reached when the motor reaches the end of travel for a move with -ve distance/velocity (CCW).

type field takes the following values:

1	Limits normally closed (default setting)
0	Limits normally open

mode field takes the following values:

0	Stop motion when a limit is hit and abort the program, then go to a predefined fault sequence, otherwise stop (default setting).
1	Stop motion when a limit is hit but continue the program. In certain applications this allows the limit switch to define a home position.

The optional ***LD*** parameter sets the required deceleration rate after hitting a limit, the default deceleration is 200 rps^2 . Changing this value will alter the **KILL** deceleration rate.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
3LIMITS(0,1,0,100)      ;both limits enabled, normally closed switch
                        ;stop motion when hit
                        ; deceleration 100rps^2
3LIMITS(1,1,0,100)      ; +ve limit disabled, normally closed switch
                        ; stop motion when hit
                        ; deceleration 100rps^2
```

To report the current configuration of the limits, type..... **3LIMITS**

The response, using the above example..... **LM1 TP1 M0 LD100.0*

See also **IS AD KILL**

Notes The default value of LIMITS type field is 1, that is normally closed.

Hitting a limit stops motion, which cannot be re-started until you drive the load back off the limit switch. An exception to this occurs while following. When following is enabled the indexer only checks to determine if the load is on a limit or not.

LIST List user program

Syntax **aLIST(*label*)**

Description The LIST command is used to view a user subroutine program in memory specified by the **label** parameter.

Properties Immediate or buffered, can't be used in labelled block, not saved by SV

Example Providing a program has been entered, typing **1LIST(ALL)** would produce the following :

1START:
1PROFILE1(360,360,400000,20)
1PROFILE2(360,360,400000,45)
1GOTO(MOVE1)
1END
1MOVE1:
1USE(1)
1G
1H
1END

Note Typing **1LIST(MOVE1)** will only produce the code following label MOVE1 as far as END. That is :

1MOVE1:
1USE(1)
1G
1H
1END

LOADENC LoadEnc settings

Syntax **aLOADENCon/off**

Description This command allows a user to specify distances in load movement steps when used with a load mounted encoder connected to X2. It can be used in combination with position maintenance or any application where you need to specify distance in load mounted encoder steps, such as feed-to-length.

The parameters used are:

on/off enables/disables load mounted encoder as the position loop feedback device 1 = ON, 0 = OFF.

When LOADENC is off, distance, velocity and acceleration are in motor steps. When LOADENC is on, distance, velocity and acceleration are in load feedback steps set by the system variable EM, Hence:

D1 = 1 load encoder step.
V1 = EM counts per sec.
A1 = EM counts per sec^2.

EM is defined as the number of load mounted encoder steps per motor revolution. The sign of EM dictates the direction of the LOADENC count. If +ve, then drive expects A leads B when motor rotates Clock Wise (CW) (+ve demanded motion).

Properties Immediate, may be included in a labelled block, saved by SV

Note The drive must be de-energised before issuing the command, you will receive **E* if issued when the motor is energised.

LOOP Repeat user code

Syntax **aLOOP(label,cycles)**

Description The **LOOP** command repeatedly calls a **labelled** block of code a number of times specified by the **cycles** parameter, the range being 0 to 65000. Note: If the number of cycles is set to 0 the loop will continue indefinitely.

Nesting of loops up to 5 levels is permitted.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example Run the grip code for a mechanical elasticity tester 6 times, and delay for 1 second between each grip cycle to allows a sensor to measure deflection.

```
2START:                 ; signifies this is the power on sequence
2DECLARE(GRIP)
2PROFILE2(150,200,4800,45)
2USE(2)                 ; Use motion parameters from profile 2
2LOOP(GRIP,6)           ; repeat the grip/ungrip code 6 times
2END                    ;

2GRIP:
2G                      ; do the move on axis 2
2O(XX1)                 ; signal grip cycle
2T1                     ; delay for one second
2H                      ; change to ungrip/grip
2G                      ; do the move again
2H                      ; change direction
2O(XX0)                 ; signal end grip cycle
2END
```

Note Also see the **EXIT**, **KILL** and **GOSUB** command.

If you use a **GOTO** command within a **LOOP**, it will stop program execution of the loop and the number of nesting levels will be set to zero.

If you exceed the number of nesting levels the program will halt and return a **E*. R(UF) will return a 'Program nesting overflow' message.

LSEL Label Select

Syntax **aLSELon/off(code,inputs,execution,*type*)**

Description The label select command allows the code following a label having the name Lnn (where nn is the detected input code) to be performed when a certain user input pattern is detected on a number of inputs in the range 1 to 5. The code can be continuously repeated or may be re-triggered, depending upon the command's execution parameter setting. The optional parameter 'type' specifies which inputs are to be tested; internal or fieldbus inputs.

The controller supports up to 16 user-defined subroutine labels, requiring 5 inputs to be able to select any 1 of the 16. You must declare each label before you use it. For example:

```
1DECLARE(L1)
1L1:
code goes here
1END
```

16 subroutine labels may be numbered in the range L1 to L31 if they are to be executed using the LSEL command.

IMPORTANT: The drive will need to use all of its inputs to select from the complete range of 16 labels unless a field-bus input-module is used.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Parameter	Range
On	1
Off	0 (default setting)
Code	0 BCD 1 binary (default setting)
Inputs	1 to 5 (default of 5)
Execution	0 continuously repeated (default setting) 1 re-triggered
Type (optional)	Reserved

The range of input code patterns is given below. Selecting a BCD code restricts the number of input codes detected (1 to 9 and 11 to 19).

Inputs					Code type/Execution label	
1	**2**	**3**	**4**	**5**	**BCD code**	**Binary code**
0	0	0	0	0	-	-
1	0	0	0	0	1	1
0	1	0	0	0	2	2
1	1	0	0	0	3	3
0	0	1	0	0	4	4
1	0	1	0	0	5	5
0	1	1	0	0	6	6
1	1	1	0	0	7	7
0	0	0	1	0	8	8
1	0	0	1	0	9	9
0	1	0	1	0	-	10
1	1	0	1	0	-	11
0	0	1	1	0	-	12
1	0	1	1	0	-	13
0	1	1	1	0	-	14
1	1	1	1	0	-	15
0	0	0	0	1	-	16
1	0	0	0	1	11	17
0	1	0	0	1	12	18
1	1	0	0	1	13	19
0	0	1	0	1	14	20
1	0	1	0	1	15	21
0	1	1	0	1	16	22
1	1	1	0	1	17	23
0	0	0	1	1	18	24
1	0	0	1	1	19	25
0	1	0	1	1	-	26
1	1	0	1	1	-	27
0	0	1	1	1	-	28
1	0	1	1	1	-	29
0	1	1	1	1	-	30
1	1	1	1	1	-	31

Example The main code configures the label select command to detect a binary code on 5 inputs (all high gives decimal 31), and if detected, to continuously run the code at label L31.

Use binary mode for PLC control and BCD for control via a thumwheel.

```
1START:
1DECLARE(L31)
1LSEL1(1,5,0)
1ARM1
.
1L31:
1A10
1V10
1G
1END
```

To check the current mode, type..... **1LSEL**

The response will be **AR1 B/D1 IN5 C/R0*

Note If the inputs remain high, the code following label 31 will run continuously. If the inactive execution mode was selected, all inputs would need to go to 00000 and then 11111 before running the code at label 31.

The selected routine will only run if no other routine is already executing.

The number of inputs available for use by LSEL depends upon the use of limits, home or registration within an application.

M Mode

Syntax	Units	Range of 'n'	Default	See also
aMn	-	see below	-	FRATE

Description The mode command sets up the mode of operation of the controller.

The values of n are:
- A – indexed move with absolute positioning
- B – continuous move with velocity controlled from the analogue input
- C – continuous move
- I – indexed move with incremental positioning
- P – step and direction base drive mode

Mode absolute – all move distances are referenced to absolute distance.

Mode bidirectional - the motor moves continuously at the velocity determined by the analogue input level. +10V gives maximum +ve velocity (determined by V), -10V gives maximum –ve velocity (determined by V), 0V gives nominal zero velocity (see also AB variable to set analogue deadband, A0 to set analogue offset).

Mode continuous –the motor moves continuously at the programmed velocity until stopped.

Mode incremental – all move distances are referenced to the starting position of each move.

Mode Position – Like 1:1 following except that the enable input becomes a true energise input. Acts like a base drive. No preset motion is allowed, but user programs will run.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example 1 The code below sets up an absolute move.

```
3W(PA,0)      ;set PA, PT, PF & PE to zero
3MA           ;mode absolute
3D1000        ;set distance
3G            ;move to absolute position 1000
3D100         ;set distance
3G            ;move to absolute position 100
3R(PT)        ;report target position
*100
```

To check the current mode, type **3M**
The response will be..... **MA*

Summary of microstepper modes

Mode	**Source**	**Enable/Energise**	**Limits**
MA	TG	Enable	Local
MB	ANA I/P	Enable	Local
MC	TG	Enable	Local
MI	TG	Enable	Local
MP	ENC I/P	Energise/Shutdown	Remote

Key:

TG – Trajectory Generator (internal command reference)

ANA I/P – ANA1+, ANA1- analogue input on X4. See also AO and AB.

ENC I/P – The encoder inputs on X4

Enable/Energise – describes the function of X4 pin 11. For a ViX indexer drive this pin is used for the enable function and its active sense is programmed using ES. Disabling the drive causes a drive fault. In mode position, the drive acts as a base drive (non indexer mode) and X4 pin 11 can be used to energise/shutdown the drive without generating a drive fault.

The MP mode provides base drive functionality, that is, step, direction, energise input and fault output. In this mode, X4 pin 11 is used to energise/de-energise the drive and the ON/OFF commands are prohibited. Local limits must be disabled by the user and monitored instead by the system controller providing the step, direction and energise signals, hence the reference to remote limits in the above table.

Example of step and direction configuration:

```
1LIMITS(3,0,0)    ;do not use local limits
1W(ES,0)          ;X4.11 low = energise, high (open circuit) = shutdown
1W(EI,0)          ;step and direction input mode
1MP               ;mode position
```

Note Status bit 25 indicates motion direction:

1=negative, CCW

0=positive, CW

In **MA** the command **H** is ignored

You should not use following in mode **MA**.

Applying a step waveform to the analogue input will cause the velocity to ramp with acceleration A to avoid stalling the stepper.

Do not use Go Home in MB mode.

Hitting a limit in MB or MC mode gives the same response.

MOTOR Motor Settings

Syntax **aMOTOR(Type,Current,Resolution,Max_vel,%thirdharmonic,Resistance, Inductance)**

Description This command describes the characteristics of the motor being used to the rest of the drive. The parameters used are:

Type – 0 to 1023 number code
Current – RMS continuous motor current (0.1 to 5.6 A in 0.1 A increments for ViX500).
Resolution – Any value from 200 to 51200 steps per rev.
Max_vel 1 to 3,000 rpm
Third Harmonic – % of third harmonic applied to current sine wave. Used to increase slow speed smoothness. Range: +/-15%.
Resistance – Winding resistance in Ohms*
Inductance – Winding inductance in mH*

*Measured line-to-line across the motor terminals.

The command sets all of the motor parameters and then calculates the optimum settings for the digital torque amplifier.

If there is no HV present when the motor command is issued, the HV is assumed to be 80V, and this figure is used for the calculation of the digital torque amplifier optimum settings. When operating the drive at a voltage other than 80V DC, make sure the HV is present when issuing the motor command. Otherwise, the settings of the digital torque amplifier will not be optimised.

IT IS IMPORTANT TO RE-ISSUE THE MOTOR COMMAND IF YOU CHANGE THE HV.

ANY CHANGES TO THE MOTOR TYPE MUST BE FOLLOWED BY A SAVE (SV) AND RESET (Z) OR CYCLING POWER TO THE DRIVE.

The motor type is stored as 10 binary digits, as follows:

bits 0 to 7	motor identification code
bit 8	temperature sensor fitted
bit 9	parallel or series connection ‘1’ = series, ‘0’ = parallel

Properties Immediate or buffered, can be used in labelled block, saved by SV

Note

[1] The motor command can take up to 12 seconds to finish execution.

[2] When changing motor type the fault and status information may not be valid until the motor has been defined, saved and the drive re-initialised.
[3] For less than 30% drive rating the motor current resolution is degraded. For 25% drive rating the maximum resolution is 50800. For 10% drive rating the maximum resolution is 26450. Whilst operating at these degraded current levels the drive and controller can still be commanded to a resolution of 51200 steps per rev.

O Output

Syntax	Units	pattern	Default	See also
aO(pattern)	-	see below	000	IS

Description The **O** command applies the specified binary pattern to the user outputs.

Pattern takes the bit values 0, 1, X, where 0 is output off, 1 is output on and X represents an unchanged state.
Pattern is 3 bits in length in the order of outputs 1 to 3
Trailing X characters are not required.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
2O(110)      ; sets outputs 1,2, ON and 3 OFF
2O(X0X)      ; leaves outputs 1,3 as they were and
             ; turns output 2 OFF
2O(101)      ;sets outputs 1 & 3 ON and turns output 2 OFF
```

OFF Shutdown motor power

Syntax	Units	Range of 'n'	Default	See also
aOFF	-	-	OFF	ON

Description Issuing an **OFF** command de-energises the drive to shutdown the motor power. The controller responds to move commands that are issued after an OFF with *E. If you check the fault variable UF, you will see the 'Drive disabled' bit set to indicate that the drive was de-energised when a move was attempted.
OFF reduces motor heating and allows manual positioning of the load, assuming the system mechanics allow this and it is safe to do so.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
1OFF    ;shut down motor power on axis 1
```

ON Turn ON motor power

Syntax	Units	Range of 'n'	Default	See also
aON	-	-	-	OFF

Description Issuing an **ON** command energises the drive and clears the current state of the drive fault registers.

The command allows execution of moves provided the motor is not on a limit.

ON will clear the User Fault and Drive Fault variable to all zeros, but if a fault is still present the motor will not energise and the fault variables will be updated once again.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
1START          ;program start-up routine
1ON             ;energise motor
.               ;attempts to clear any faults
.               ;
1END            ;

1FAULT          ;fault handling routine
1"FLT"          ;send warning over comms.
1TR(IN,=,XX1)   ;wait for 'reset' input 3
1GOTO(START)    ;re-run start routine
1END            ;
```

This small section of program shows the use of the ON command at the start of the code and the use of a fault routine to attempt a program re-start if a fault occurs.

Note Issuing an ON command will clear all user status flags.

POSMAIN Position maintenance

Syntax **aPOSMAINon/off(dead_band_range,*output*,*settletime*)**

Description **POSMAIN** must use an encoder connected to X2 to monitor the system's actual position. The command enables position maintenance. The **on/off** parameter takes the values of 1 for ON and 0 for OFF. With position maintenance enabled, if the motor shaft stops within the **dead band range** for the optionally specified settle time then the indexer will regard the move as having been completed. If the motor shaft is stationary outside of the dead band range, position maintenance will attempt to move the motor shaft to its target position.

Status bit 17 will be set while executing a position maintenance move.

The optional parameter ***output*** specifies a user output that will be latched ON when a corrective move is made. The output will be turned OFF when the next G (go), GH (go home), any ON command or Z (reset) command is received, and status bit 17 will be reset.

The optional parameter ***settletime*** specifies how long in milliseconds that the indexer will wait after motion has ceased, before checking the feedback encoder. Motion has ceased when the motor has been commanded to stop and the in-position timer has timed out. The in-position signal will only be set (high) after the in-position time and the settle-times have lapsed and the motor shaft has been positioned within the deadband.

For more information on POSMAIN refer to the ***Position maintenance*** section in ***Control of ViX Drives***.

Parameter ranges are:

Parameter	Range	Units	Defaults
On/off	1 ON, 0 OFF	refer to table of distance units for enabled commands in section 4.	0
Deadband Range	0 to 32767		10
Output	0 to 3 (0 = no output)		0
Settle time	0 to 65535	milliseconds	0

Note: dead band range is measured in encoder count units as a +/- band, consequently a dead band of 10 will be equivalent to ±10 or 20 units wide.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
1POSMAIN1(10,3)     ; Enable position maintenance. Allow
                    ; a 10 step dead band window and set o/p 3
                    ; when within the desired range
1POSMAIN1(20)       ; Enable position maintenance.  Allow a
                    ; 20 step dead band window.  Output not set.
```

Note

[1] Position maintenance is performed at the end of a move to correct any overall position error.

[2] If encoder resolution is different to the motor resolution, then you must set system variable EM and enable LOADENC. The deadband range will work in EM* steps. If setting EM +ve, the encoder expects A to lead B for +ve commanded motion.

[3] The error window is measured in motor steps with LOADENC and SCALE disabled, load steps with LOADENC enabled, and user steps with SCALE enabled.

[4] If scaling is enabled, the value given in the deadband will remain the same, but will now be in user units.

[5] Use encoder port X2 (primary encoder).

[6] With POSMAIN enabled DO NOT USE FOLLOWING.

[7] Position maintenance parameters can only operate correctly after MOTOR, LOADENC, SCALE and the EM variable have been correctly configured.

[8] POSMAIN velocity is fixed as V1. The acceleration rate is the one used in the last GO command.

[9] Position maintenance mode can be selected and de-selected during the execution of a program.

*EM = encoder counts per rev (4 X line count).

PROFILE Define move profile

Syntax **aPROFILEnumber(AA,AD,D,V)**

Description The **PROFILE** command sets up a table of move profiles in the controller memory. These profiles can be recalled at any time by the **USE** command.

The **PROFILE** command parameters are:

Acceleration	**AA**
Deceleration	**AD**
Distance	**D**
Velocity	**V**

Ranges for the **AA**, **AD**, **D** and **V** commands are as stated for each individual command.
The range of PROFILE number is 0 to 8, but PROFILE0 cannot be defined.

Use PROFILE0 to read the current profile settings. The format of the returned message will be:

**0 AA10.0 AD10.0 D4000 V1.00*

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example Profile 1 is to represent a move of 1500 steps on axis 3 at a velocity of 25 rps and acceleration/deceleration of 200 rps^2:

3PROFILE1(200,200,1500,25)

Profile 2 is to represent a move of 4800 steps on axis 3 at a velocity of 45 rps, acceleration of 150 rps^2 deceleration of 200 rps^2:

3PROFILE2(150,200,4800,45)

The following move profiles will now be available in memory on axis 3:

Profile number	1	2
Acceleration	200	150
Deceleration	200	200
Distance	1500	4800
Velocity	25	45

The move parameters specified by Profile 2 may be used (that is, copied to profile 0) with the statement..... **4USE(2)**

Note A profile command will overwrite any individually programmed values of acceleration, deceleration, distance and velocity once the USE command is issued.

PS Pause

Syntax	Units	Range of 'n'	Default	See also
aPS	-	-	-	C

Description The **PS** (pause) command causes immediate command execution to cease until a **C** (continue) command is issued. The command is useful as a debug aid when testing small trial code blocks.
The PS command cannot be used whilst running a program.

Properties Immediate or buffered, can't be used in labelled block, not saved by SV

Example

```
0PS       ;global pause
1D4000    ;setup axis 1
1V5       ;  .
1A50      ;  .
2D8000    ;setup axis 2
2V10      ;  .
2A100     ;  .
0G        ;global GO
0C        ;global continue
```

Note If the input command buffer is filled during a pause *E will be reported (assuming EX is set to speak whenever), and the status LED will continually flash red then green. To clear this condition cycle the power.

R Report system parameter

Syntax	Units	Range of 'n'	Default	See also
aR(system_variable)	-	-	-	W

Description The **R** command allows the user to read the specified **system variable**.

Properties Immediate or buffered, can be used in labelled block, not saved by SV
Note: aR(RB) is immediate only

Example

```
2R(AO)          ;report the current value of variable AO
```

The response could be **1500*

REG Registration move

Syntax **aREGon/off(edge,profile_number,*hold_off_distance*,*registration window,output*)**

Description The **REG** command, once turned ON, defines a registration move. After a number of steps, determined by the optional **hold off distance**, the controller will begin to search for a valid registration signal.

Once a valid registration mark has been detected the registration move is performed using the move parameters taken from the previously defined profile* (profile_number in the command parameters). At the end of the registration move the user program GOSUBs to the code immediately following the REG label. If no registration mark is detected, the standard move profile completes and the user program GOSUBs to the code immediately following the NOREG label.

* Registration will always occur in the current move direction. If the direction in the defined profile is different to the current move direction, the direction information in the defined profile is ignored.

An optional output can be programmed to indicate that a move that has been armed is ready for registration. This would normally be after the move has started or after the hold-off distance (if defined). The output chosen must be within the range of allowable outputs (1 to 3). The default value is no output.

Once registration has been setup, it can be applied (turned ON) using the simplified form of command:

aREG1 or turned OFF using: **aREG0**

Properties Immediate or buffered, can be used in labelled block, saved by SV

Parameter	Range	Units	Comments
On/Off	1 or 0 (default)		1 ON, 0 OFF
Edge	1 or 0 (default)		1 rising, 0 falling
Profile number	1 to 8		Must be user defined
Hold off distance	0 to 2147483647	steps	default 0
Registration window	0 to 2147483647	steps	default 0
Output	0 to 3		Default is no output (0)

Example

```
2START:
2PROFILE1(10,10,40000,5)
2PROFILE2(20,20,20000,10)
2REG1(1,1,10000)
2USE(2)
2G
2END

2REG:
2O(XX1)          ; Turn output 3 on : increment batch counter
2T0.5            ; Delay execution for 500mS
2O(XX0)          ; Turn off output 3
2END

2NOREG:          ; if we come here we didn't have a valid reg mark
2O(X1X)          ; Turn op2 on :push unlabelled product off conveyor
2T0.25           ; Delay execution for 250mS
2O(X0X)          ; Turn off output 2
2END
```

RFS Return to factory settings

Syntax	Units	Range of 'n'	Default	See also
aRFS	-	-	-	SV

Description Issuing an **RFS** command initialises the controller to factory default settings. The drive must be de-energised (OFF) for **RFS** to be executed.
Factory settings must be saved using the SV command before they take effect.

The default settings are:

All labels cleared, all outputs set to logic low.

Note: the RFS state of ARM is start disabled, fault enabled (**ARM01**).

Properties Immediate, can't be used in labelled block, saved by SV

Example

```
1LIMITS(3,0,0,900)      ;define limits
1RFS                    ;return to factory settings
1SV                     ;save factory settings
1LIMITS                 ;report limits
*LM0 TP1 M0 AD 200
```

S Stop

Syntax	Units	Range of 'n'	Default	See also
aS	-	-	-	PS, K

Description Use the **S** command to bring motion to a controlled stop. The command will use the current value of deceleration as specified by either the immediate A or AD commands or the current profile being used. This command also aborts label execution when used from the command line.

Use the command from the command line or within a label.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1G      ;start the move
1S      ;stop the move
```

SCALE Scale settings

Syntax **aSCALEon/off(SCLA,SCLD,SCLV, PEU)**

Description This command allows a user to specify Acceleration, Distance and Velocity in their own chosen units.

The drive firmware needs to know how many commanded position steps there are in a user unit and then how many fractions of a unit there are for an A, D or V of 1.

The parameters used are:

SCLA – 1 to 100,000,000 Used to scale A.
A of 1 unit/s^2 = PEU encoder counts per sec^2. If SCLA <>1, all acceleration values entered are internally divided by the SCLA parameter value. Default 1.

SCLD – 1 to 100,000,000 Used to scale D.
D of 1 unit = PEU encoder counts. If SCLD <>1, all distance values entered are internally divided by the SCLD parameter value. Default 4000.

SCLV – 1 to 100,000,000 Used to scale V.
V of 1 unit/s = PEU encoder counts per sec. If SCLV<>1, all velocity values entered are internally divided by the SCLV parameter value. Default 1.

PEU – 1 to 100,000,000 **–** Position Encoder steps per Unit : defines the number of position feedback encoder steps in a user unit. The steps referred to are motor steps if LOADENC is disabled, or load mounted encoder steps if LOADENC is enabled.
Default unit = 1 motor rev (rotary) 1 pole pitch (linear)
Default PEU = 4000

Example:
Suppose we have a motor attached to a linear table. The motor resolution has been set to 4000 steps per rev. The linear table has a load mounted 10um encoder and is 0.5m long. It takes 80 motor revs to move the table its complete length. The motor has to position the table in 100 distinct linear positions at a speed range of $0.02ms^{-1}$ to $0.2ms^{-1}$.

- 50000 / 80 = 625 load encoder counts per motor revolution

Set variable EM to 625
Enable LOADENC

- 0.5m at 10um = 50000 load counts complete table length

Set PEU to 50000

- Table to move to 100 distinct locations

Set SCLD to 100. D1 = one hundredth of the table length.

- 0.5m / $0.01ms^{-1}$ = 50

Set SCLV to 50. Then V2 = $0.02ms^{-1}$ and V20 = $0.2ms^{-1}$

The scaling is applied to the next move made upon issue of the GO command.
A, AA, AD, D and V always report user units.
In any application the SCALE command should be issued once only. Its purpose is to allow for a fixed user unit to motor steps scaling. **It is not designed to be changed on the fly and unpredictable results may occur if the command is used in this manner.**

Properties Immediate, may be included in a labelled block, saved by SV

Note

- If SCALE is enabled and PEU/SCLD is a non-integer value, then the drive will return a *E.
- Stall and Posmain deadbands work in user units when SCALE is enabled.
- When scale is enabled, the value for deadband is not adjusted, but it is now in user units and may need to be changed to maintain the same displacement.
- Enabling scale does not change the values of A, D and V, but they are now measured in user units.
- If scale is turned OFF, remember to set A, D and V to appropriate values before issuing a GO command. Values of V may be out of range, unless re-issued in motor or load units.

STALL Stall detect

Syntax **aSTALLon/off(error_window,mode,*output*)**

Description The **STALL** command is used to enable stall detect. To use this command an encoder must be connected to X2 to monitor the system's actual position. The **on/off** parameter takes the values of 1 for ON and 0 for OFF and has a default value of OFF. If the error between demanded position and encoder position exceeds the **error window**, then the drive will generate a stall fault.

A stall fault will cause the ST LED on the front panel to go from green to red and the optional output (if specified) to switch to a '1'. Motion is stopped, status bit 18 is set (possible stall), drive fault bit 1 is set (composite fault) and drive fault bit 19 is set (stall condition). The drive will remove any pending commands from its input buffer, and abort any program that is running.

Setting the mode parameter to '1' will run a fault routine (if one is defined) once the motor has stopped. No further action is taken if the mode parameter is set to '0'.

The fault condition will clear when the next **G** (go), **GH** (go home), any **ON** command or **Z** (reset) command is received. The output (if specified) will revert to '0' and the ST LED will change back to green. All status and drive bits will be reset. When stalled, the optional output can be reset using the **O** command, but this will not reset any status or drive faults.

Because distance units can be changed using scaling, you must set the value of the EM system variable. If EM is positive, the encoder port X2 expects to see A leading B when the motor shaft rotates clockwise.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Parameter	Range	Units	Default
on/off	1 ON, 0 OFF	See table of distance units	0
error window	0 to 65535		4000
Mode on stall	1 run fault, 0 no fault		0
output	0 to 3 (0 = no output)		0

Example **1STALL1(100,1,3)** ; Stall detect on. Use 100 step error window
; Run fault if stall detected and turn ON o/p 3

Note

[1] The error window is measured in motor steps with LOADENC and SCALE disabled, load steps with LOADENC enabled, and user steps with SCALE enabled.
[2] If scaling is enabled, the value given in the error window will remain the same, but is now in user units.
[3] Drive status LED will turn red after a stall occurs (reset as for optional output)
[4] Position maintenance parameters can only operate correctly after MOTOR, LOADENC, SCALE and the EM variable have been correctly configured.

STATUS STATUS of Drive

Syntax	Units	Range of 'n'	Default	See also
aSTATUS	-	-	-	

Description Use this command to check the state of a drive. It is intended for set-up purposes rather than for use when a program is running.

Properties Immediate or buffered, can't be used in a labelled block, not saved by SV

Example **1STATUS** ;checking the configuration and state of a drive

```
1STATUS
*ViX500IM-Stepper  Copyright 2003 Parker-Hannifin
*Firmware: REV 2.1bD-
*Serial number: ..541935.00.1.1.CcD-ViX500IM
*Control card revision 2                Stepper drive
*Power card revision 3                  Power stage E500
*
*
*FPGA_ID (read).........   20b0         FPGA_ID (file)....... 20b0
*
*MOTOR TYPE ............  255           RESOLUTION .......... 4000
*CONT. STALL CURRENT ...    4.5 Amps
*
*
*MOTOR SUPPLY...........     79 V       AUX SUPPLY...........  5.1 V
*I/O SUPPLY.............     23 V       I/O CONFIGURATION.... 8160
*INTERNAL TEMPERATURE...     49 C       HEATSINK TEMPERATURE.   47 C
*
*INCREMENTAL INDEXING (MI)
*POS.MAIN.WINDOW...........10           STALL WINDOW..........4000
*VELOCITY (V)...........   1.00         DISTANCE (D)......... 4000
*ACCELERATION (AA)......   10.0         DECELERATION (AD).... 10.0
*CURRENT POSITION (PT)..   0            ERROR (PE)...........    0
*POSITION MODULUS (PM)     0
*
*AXIS: READY
*DRIVE FAULTS (DF): 0000_0000_0000_0000_0000_0000_0000_0000
*DRIVE STATUS (ST): 0000_0000_1000_0000_0001_0000_0000_0010
*USER  FAULTS (UF): 0000_0000_0000_0000_0000_0000_0000_0000
```

STOP STOP Input

Syntax	aSTOPon/off(mode)

Description The **STOP** Input command determines the 'stop input' functionality of input 1. When input 1 is active, IS = 1XXXX. The **on/off** parameter enables/disables the stop input taking the values 1 for ON and 0 for OFF and has a default value of OFF.

The **mode** can be set as follows:

0 Stop motion when input 1 is active (IS = 1XXXX) and abort the user program (default setting).
1 Stop motion when input 1 is active (IS = 1XXXX), but continue the user program which is able to execute further commands.

If stop input 1 is active (IS = 1XXXX), then status flag 28 (ST4.4) will be a '1'.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

```
1RUN:           ;start the move
1MC
1STOP1(1)       ;enable the stop input & program continue
1G              ;input 1 goes active during motion
1TR(IP,=,1)     ;wait for input 1 & in position settle time
1"*STOPPED"     ;*STOPPED transmitted when motion has halted
1IF(ST4,=,XXX1XXXX)
1"*INPUT=1"
1END
```

Note The stop input only stops indexed motion. It has no effect on following or step & direction (MP) modes.

SV Save configuration

Syntax	Units	Range of 'n'	Default	See also
aSV	-	-	-	Z

Description When the **SV** command is issued, the current controller system variables and user programs are stored in non volatile memory. Any data saved, will be restored following the next power-ON cycle.

The number of write/save cycles is 1 million.

Normally, there will be a delay of approximately one-second before a command following a save configuration is executed. If a program is running or is being downloaded when the SV command is issued, a delay of 10 seconds is allowed for the program to terminate or to finish downloading. After 10 seconds, if the program is still running or downloading a user fault is generated – cannot execute command, drive not ready (bit 19).

Wait 1-2 seconds before sending any other command following an SV.

Properties Immediate, can't be used in labelled block

Example

```
1RFS          ;return drive to factory settings
1A150.1       ;acceleration set to 150.1rps2
1SV           ;save current settings
1Z            ;reset drive
1A            ;report current value of acceleration
*150.1
```

T Time delay

Syntax	Units	Range of 'n'	Default	See also
aTn	seconds	0.05 to 10	none	IF

Description The **T** command pauses program execution for the time specified by the **delay** parameter. Timing resolution is to within 50ms increments. Any time value specified within the range 0.05 to 10 seconds will be rounded down to the nearest 0.05 second increment. Any value programmed outside of this range will generate an error (**E out of range*).

The receipt of an immediate command whilst executing a time delay causes the delay to end.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
4T6       ; delay for 6 seconds
4T0.38    ; delay 0.35 seconds (rounded down)
```

TR Wait for trigger

Syntax **aTR(system_variable,relation,value)**

Description The **TR** command pauses command execution until the trigger condition is met.

The trigger condition is met if the **relation** between **system_variable** and **value** is true.

Valid relations for the comparison are:

=	Equals
<>	Does not equal
>	Greater than
<	Less than

Value is a number generated by the **system_variable** being tested. Refer to the system variables table for more information.

Also see system variable Trigger Timeout (**TT**).

Refer to the table of system variables that can be used for conditional control.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example **3TR(PA,>,2000)** ; wait for position absolute to be >2000 steps

3TR(IN,=,X11XX) ; wait for user inputs 2 and 3 to be high

Notes If you wish to use the **TR** command during motion, command queuing (system variable CQ) must be set for continuous execution (CQ=0).

Issuing a K or S from the command line will clear a trigger condition.

If the input command buffer is filled whilst waiting for a trigger *E will be reported (assuming EX is set to speak whenever), and the status LED will continually flash red then green. To clear this condition, cycle the power.

USE Use

Syntax	Units	Range of 'n'	Default	See also
aUSE(profile)	-	1 to 8	-	PROFILE

Description The **USE** command copies the pre-defined profile to the current move parameters.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1PROFILE1(200,20,1500,25)      ;define profile 1
1PROFILE2(150,200,4800,45)     ;define profile 2
1USE(2)                        ;use motion profile 2
```

WARNING

If you attempt to use an undefined PROFILE, PROFILE0 is used with no error indication.

V Velocity

Syntax	Units	Range of 'n'	Default	See also
aVn	see SCALE	0.001 to 50.000	1	PROFILE SCALE

Description Velocity command **V** sets or reports the programmed velocity of the motor.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

To set the velocity of axis 3 to 25 rps, type..... **3V25**
To report the current velocity of axis 3, type..... **3V**
The controller responds with..... **25.0*
No units are reported.

Note

[1] A programmed value of velocity can be overwritten by a PROFILE command once the USE command has been issued, but subsequent values of velocity can be programmed to override the value in use.
[2] With SCALE enabled, a maximum value of 5000 (user-units) is permitted.
[3] If a value is entered that requests a velocity greater than the maximum velocity set in the MOTOR command, *E will be returned.

W Write system variable

Syntax	aW(system_variable,value)

Description The **W** command allows you to set a specified **system variable** to a particular **value**.
Refer to the table of system variables for more information.

Properties Immediate or buffered, can be used in labelled block, saved by SV

Example

Set system variable DC to 1	**2W(DC,1)**
Report the current value of system variable DC	**2R(DC)**
The controller responds with	**1*

Note See also R command.

Z Reset

Syntax	Units	Range of 'n'	Default	See also
aZ	-	-	-	SV

Description The **Z** command resets the drive's controller. This is similar to power cycling the controller. Upon restart, the user program following the START: label will execute only if the ARM command = 1X.

Wait 1-2 seconds before sending any other command following a Z.

Any commands pending before the Z is issued will be terminated and any buffers and user stacks cleared.

Properties Immediate or buffered, can't be used in labelled block, not saved by SV

Example To reset all drives, type **..... 0Z**

Set comms address remotely

Syntax	Units	Range of 'n'	Default	See also
a#n	-	1 to 255	0	-

Description This command (#) allows you to set the unit address via software. It allows addresses up to 255 to be used. Upon receipt of the command, the controller will send a #n+1 command along the daisy chain provided the echo mode is set. Once received you must send a SV command to save the address configuration.
To address a specific axis place the current address 'a' before the # symbol.

0#0 is not supported on this product.

Auto addressing can be used, for example sending **#1** to axis 1 of a 3 axis system will echo back *#4* (meaning the axes have been given addresses 1, 2 & 3). Save the address configuration of all axes using the command **0SV**.

The command can be used to specify the style of communications required, for example **1#4(232)** will set axis 1 to address 4 and specifies the use of RS232. Note this form of command is immediate and is auto saved as soon as you hit return, consequently take care. If you have no RS485 interface fitted to your PC and you issued **1#4(485)*** the drive will switch to RS485 and will auto save the change, leaving you without any means of communicating with the drive. See ***Forcing a Hardware RFS*** in the ***Maintenance & Troubleshooting*** section.

Auto addressing can be used with style of communications type commands, allowing such commands as #n(232) or #n(485) where n is the primary axis you wish addressing to start from.

*You must have a RS485 drive module fitted for the drive to recognise the command.

Properties Immediate or buffered, can't be used in labelled block, saved by SV

Example If a system with axes 1,3,7,2 (in that order) is sent the command **#10**, the axes will become 10, 11, 12, 13. Note that the #10 will not be displayed on your PC screen, but you will receive the response #14 after pressing the enter key.

For multi-axis systems using RS232, auto addressing can be used when the drives are interconnected via the RJ45 connectors (X6 & X7). Primary communication needs to be via the front panel D-type X3 connector.

“ “ Quote command

Syntax	Units	Range of ‘n’	Default	See also
a“ “	-	-	-	-

Description Use QUOTE to send messages to other drives or displays. Using the RS232 link, up to 20 ASCII characters are available to transmit the required command or message in exactly the way in which it was entered. Only ASCII characters between decimal 32 (space character) and decimal 126 (tilde ‘~’ character) inclusive are allowed.

Properties Immediate or buffered, can be used in labelled block, not saved by SV

Example

```
1DECLARE(EXAMP)     ;declare label
1EXAMP:             ;label EXAMP
1”*TEST”            ;quote TEST
1END                ;end label
1GOTO(EXAMP)        ;goto label EXAMP
*TEST               ;output message
```

Note To speed-up communications when addressing a number of drives, precede the quoted text with an asterisk ‘*’. All other axes, apart from the one being addressed, will ignore the quoted text and this will save processing time.

The command can be used to debug routines that do not appear to run. Add a quote command to the suspect portion of code and see if it appears when the code is executed.

Use quote commands sparingly as they can use a lot of available program memory.

System Variables

Var	Name	R	W	Range/default value
AB	Analogue Deadband	Y	Y	0 to +255, default = 0
AI	Analogue Input	Y	N	-2047 to +2047
AO	Analogue Offset	Y	Y	-2047 to +2047, default = 0
BR	BAUD rate	Y	Y	9600 or 19200 bits per second (9600 default)
BU	Buffer usage	Y	N	0 to 100% of program buffer used
CQ	Command queuing	Y	Y	1= Pauses until move complete (default) 0= continuous execution
DC	Damping Configuration	Y	Y	0 = settling time damping OFF (default) 1 = settling time damping ON
DF	Drive Fault status	Y	N	See below:
DF1	Drive Fault status	Y	N	First byte of 32-bit DF variable
DF2	Drive Fault status	Y	N	Second byte of 32-bit DF variable
DF3	Drive Fault status	Y	N	Third byte of 32-bit DF variable
DF4	Drive Fault status	Y	N	Fourth byte of 32-bit DF variable
EI	Encoder Input	Y	Y	0=step/dir, 1=cw/ccw, 2=quad ABZ, de-energise drive to change
EM	Encoder count per rev.	Y	Y	1 to 4200000 (default 4000)
EO	Encoder signal Output	Y	Y	0=step/dir, 1=cw/ccw, 2=quad ABZ, de-energise drive to change
EQ	Echo Queuing	Y	Y	0=normal, 1=wait for <CR>, 2=cmd response only
ES	Energise Sense	Y	Y	Sets the sense of the external enable/enable_bar signal 0=low signal to enable 1=high signal to enable
EX	Comms. Response Style & Echo Control & Physical Interface (RS232)	Y	Y	0= speak when spoken to, echo off, default for RS485 1= speak whenever, echo off 2= speak when spoken to, echo on 3= speak whenever, echo on, default for RS232

Var	Name	R	W	Range/default value
FB	Fieldbus Baud			Refer to CANopen user guide
FC	Fieldbus Control			Refer to CANopen user guide
FN	Fieldbus Node ID			Refer to CANopen user guide
FP	Fieldbus Protocol	Y	Y	Refer to CANopen user guide
HF	Home Final velocity	Y	Y	Sets the final velocity of the home move Range: 0.001 to 5.0 rps (default 0.1)
IC	Input/Output Configuration	Y	Y	Input pull-up/down, output source/sink configuration 0 to 8191 default:8160
IN	Inputs (on drive)	N	N	Local drive inputs 1 to 5, same format as IS command
INn	Inputs (expansion)	N	N	Fieldbus expansion inputs, IN1=bank1, IN2=bank2.
IP	In Position flag	Y	N	1= In position or 0= not yet in position
IT	In Position Time	Y	Y	1 to 500mS, default=10mS
MS	Motor Standby	Y	Y	Range 10% to 100% of programmed current (default 50%)
MV	Moving	Y	N	Flag 1= moving or 0 = not moving
PA	Position Actual	Y	N*	-2,147,483,648 to 0 to 2,147,483,647
PE	Position Error	Y	N*	+/- 65535
PF	Position Following	Y	Y	-2,147,483,648 to 0 to 2,147,483,647
PI	Position Incremental	Y	Y	-2,147,483,648 to 0 to 2,147,483,647
PM	Position Master	Y	Y	-2,147,483,648 to 0 to 2,147,483,647 Note: a write to PM sets the modulus
PR	Position Registration	Y	N	The primary (X2) feedback position (PA) on the last active transition on input 2 (start of valid REG move). Range: -2,147,483,648 to 0 to 2,147,483,647
PS	Position Secondary	Y	N	The PM count position on the last active transition on input 1 (falling edge viewed using IS). Range: -2,147,483,648 to 0 to 2,147,483,647
PT	Position Target	Y	Y	-2,147,483,648 to 0 to 2,147,483,647 Trajectory generator open loop target position
RB	Ready/Busy flag	Y	N	Flag 0= ready or 1= busy
RM	Registration Move	Y	N	Flag 1= reg move in progress 0 = not doing reg move
RV	ReVision of software	Y	N	x.yy major.minor
SC	S Curve configuration	Y	Y	0 = S curve accel/decel disabled (default) 1 = S curve accel/decel enabled
SN	Serial number	Y	N	reserved

Var	Name	R	W	Range/default value
ST	Status of indexing	Y	N	See below
ST1	Status of indexing	Y	N	First byte of 32-bit ST variable
ST2	Status of indexing	Y	N	Second byte of 32-bit ST variable
ST3	Status of indexing	Y	N	Third byte of 32-bit ST variable
ST4	Status of indexing	Y	N	Fourth byte of 32-bit ST variable
TT	Trigger Timeout	Y	Y	Optional timeout for trigger command 0-65 seconds in 0.01 increments. User status bit 8 is set to indicate timeout occurred before trigger condition met. Bit is clear if trigger condition met before timeout. The default time is = 0.00 (no timeout).
UF	User program Fault status	Y	N	See below
UF1	User Fault Status	Y	N	First byte of 32-bit User Fault status word
UF2	User Fault Status	Y	N	Second byte of 32-bit User Fault status word
UF3	User Fault Status	Y	N	Third byte of 32-bit User Fault status word
UF4	User Fault Status	Y	N	Fourth byte of 32-bit User Fault status word

*Can be set to 0 only.

Drive Faults

Bit	Bit Tested	Stop	Type	DF Information
1	DF 1.1			Composite fault
2	DF 1.2	K	T	+/-15V supply rail
3	DF 1.3	K	R	Motor HV under-voltage trip point reached
4	DF 1.4	K	R	Motor HV over-voltage trip point reached
5	DF 1.5			
6	DF 1.6	CD	R	Vio over-voltage trip point reached
7	DF 1.7	K	T	Encoder / Auxiliary 5V under voltage trip
8	DF 1.8	K	SLEEP	Impending power loss (24V – logic supply)
9	DF 2.1			Reserved
10	DF 2.2			Reserved
11	DF 2.3	CD	R	Motor over temperature
12	DF 2.4	CD	R	Ambient over temperature
13	DF 2.5	CD	R	Drive over temperature
14	DF 2.6	K	T	Incompatible firmware version
15	DF 2.7	K	T	Unrecognised power stage
16	DF 2.8	K	T	Controller diagnostic failure
17	DF 3.1	K	R	Output stage over current
18	DF 3.2	CD	R	Output driver over current
19	DF 3.3	C	R	Tracking limit exceeded : Stall condition
20	DF 3.4			Reserved
21	DF 3.5	CD	R	Drive disabled – check enable input and state of ES variable
22-24	DF 3.6/8			Reserved
25	DF 4.1	K	T	Watchdog 1
26	DF 4.2	K	T	Watchdog 2
27	DF 4.3	K	T	Watchdog 3
28-31	DF 4.4/7			Reserved
32	DF 4.8			CAN I/O errors

Key:
C : Performs controlled stop.
CD : Controlled stop then de-energise
K : Performs motion kill – quick stop. Possible instant de-energise depending on fault source.
R : Recoverable without power cycle
SLEEP : Drive shuts down completely – no comms, requires power-cycle to recover
T : Terminal (requires power cycle or repair before drive will energise / operate once again)

Status Bits

Bit Number	Bit Tested	Status Information
1	ST1.1	Command processing paused
2	ST1.2	Looping (command executing)
3	ST1.3	Wait for trigger (input)
4	ST1.4	Running program
5	ST1.5	Going home
6	ST1.6	Waiting for delay timeout
7	ST1.7	Registration in progress
8	ST1.8	Last trigger command timed out
9	ST2.1	Motor energised
11	ST2.3	Event triggered - active until trigger inputs are reset
12	ST2.4	Input in LSEL not matching label
13	ST2.5	-ve limit seen during last move
14	ST2.6	+ve limit seen during last move
16	ST2.8	Reserved
17	ST3.1	Executing a position maintenance move
18	ST3.2	Possible stall
19	ST3.3	Moving (in motion)
20	ST3.4	Stationary (in position)
21	ST3.5	No registration signal seen in registration window
22	ST3.6	Cannot stop within the defined registration distance
23	ST3.7	Reserved
24	ST3.8	Reserved
25	ST4.1	In motion, 0 for positive motion, 1 for negative motion
26	ST4.2	Reserved
27	ST4.3	Following enabled = 1, not following = 0
28	ST4.4	STOP input active
29	ST4.5	Load mounted encoder enabled
30	ST4.6	Scaling enabled
31	ST4.7	Command input inverted

User Faults

Bit Number	Bit Tested	UF Information
1	UF1.1	Value is out of range
2	UF1.2	Incorrect command syntax
3	UF1.3	Last label already in use
4	UF1.4	Label of this name not defined
5	UF1.5	Missing Z pulse when homing
6	UF1.6	Homing failed - no signal detected
7	UF1.7	Home signal too narrow
8	UF1.8	Drive de-energised
9	UF2.1	Cannot relate END statement to a label
10	UF2.2	Program memory buffer full*
11	UF2.3	No more motion profiles available
12	UF2.4	No more sequence labels available
13	UF2.5	End of travel limit hit
14	UF2.6	Still moving
15	UF2.7	Deceleration error
16	UF2.8	Transmit buffer overflow
17	UF3.1	User program nesting overflow
18	UF3.2	Cannot use an undefined profile
19	UF3.3	Drive not ready
22	UF3.6	Save error
23	UF3.7	Command not supported by this product
24	UF3.8	Fieldbus error
25	UF4.1	Input buffer overflow
26 to 32	UF4.2/8	Reserved

*sends an ASCII ‘bell’ character to indicate a buffer overflow condition.

Command List

Command	Description
A	Acceleration/Deceleration
AA	Acceleration
AD	Deceleration
ARM	Enable event triggered code
C	Continue
CLEAR	Clear user code
D	Distance
DECLARE	Declare
E	Enable/disable communications
EXIT	Exit from loop
FOLLOW	Configure following
FRATE	Feed rate override
G	Go
GH	Go home
GOSUB	Go to subroutine
GOTO	Go to routine
H	Change direction
HOME	Configure homing
IF	Test condition
IS	Input status
K	Kill
LIMITS	Configure limit inputs
LIST	List user program
LOADENC	LoadEnc settings
LOOP	Repeat user code
LSEL	Label select
M	Mode
MOTOR	Motor settings
O	Output
OFF	Shutdown motor power
ON	Turn on motor power

Command	Description
POSMAIN	Position maintenance
PROFILE	Define move profile
PS	Pause
R	Report system parameter
REG	Registration move
RFS	Return to factory settings
S	Stop
SCALE	Scale settings
STALL	Stall detect
STATUS	Report status of drive
STOP	Stop input
SV	Save configuration
T	Time delay
TR	Wait for trigger
USE	Use
V	Velocity
W	Write system variable
Z	Reset
#	Set comms address remotely
“ ”	Quote command

7. ViX Maintenance and Troubleshooting

Maintenance

ViX drive systems do not require any routine maintenance, but occasional checking of the following points is recommended.

Motor inspection

Periodically check the motor to ensure that the mounting bolts and couplings are tight. Check that the motor cables are not being damaged by moving parts and are not being pulled or forced into tight bends during machine operation. Check all cable connectors and particularly the safety earth connection.

Drive inspection

Check that the drives are clear of loose material and that there is adequate clearance to allow a free flow of air through the ventilation slots. Check that drive fixings are tight and that the motor screen connection is secure.

Troubleshooting

IMPORTANT NOTE - ensure that power is turned off before any connections are removed or changed. **Removing a drive with power applied can turn a recoverable fault situation into a major problem.**

WARNING – Risk of damage and/or personal injury

The ViX drives described in this user guide contain no user-serviceable parts. Attempting to open the case of any unit, or to replace any internal component, may result in damage to the unit and/or personal injury. This may also void the warranty.

Communication Problems

When attempting a Connect from the Terminal menu, if the connection fails with the following error message:

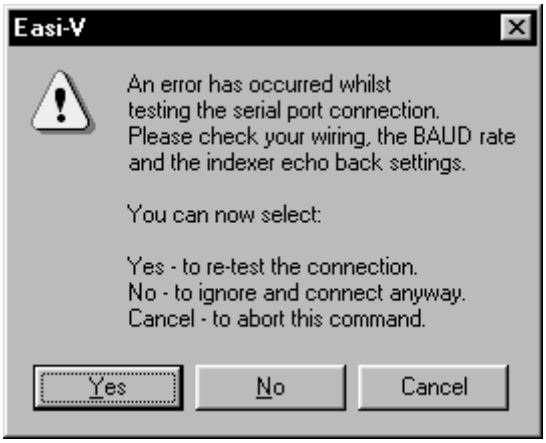


Figure 7-1. Communications Failure Error Message

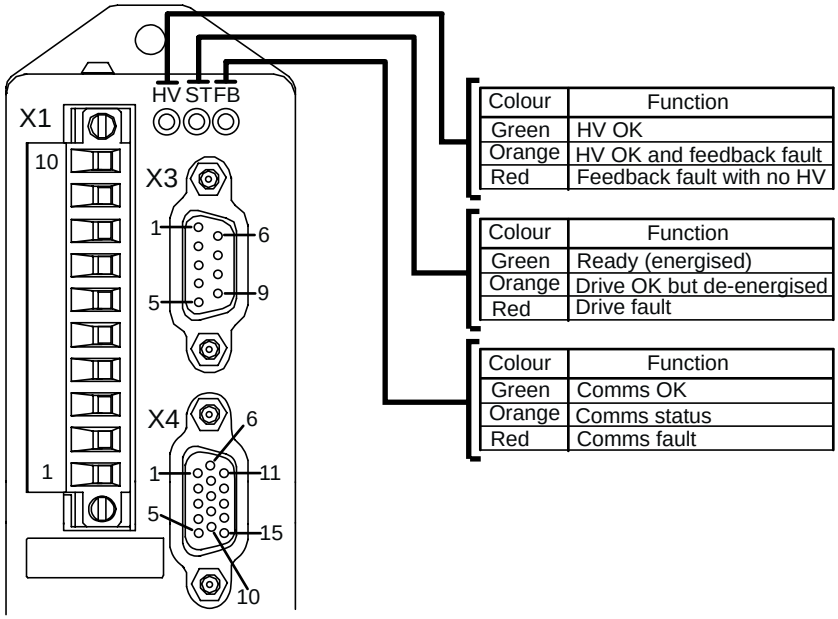
Check the following:

1. Ensure the serial port configuration is set correctly in EASI-V and you select the correct serial COM port.

2. An RS232 communications link can be loop tested by removing the communications D-type plug where it connects to the drive and placing a short between pins 4 and 5. In this condition, any command sent from the terminal window should be echoed back, confirming the integrity of the overall RS232 link. If this does not happen, check the RS232 lead connections and the PC serial port.

Note: wiring of the RS232 lead must conform to that recommended in the Hardware Installation section, a null modem cable cannot be used.

Drive LED Indicators



Colour	Function
Green	HV OK
Orange	HV OK and feedback fault
Red	Feedback fault with no HV

Colour	Function
Green	Ready (energised)
Orange	Drive OK but de-energised
Red	Drive fault

Colour	Function
Green	Comms OK
Orange	Comms status
Red	Comms fault

Figure 7-2. Drive LED Indicators

Complete LED Diagnostics

An EASI-V version of this table is available for quick on-line viewing.

LED	Colour(s)	Flash rate	Functional description
HV	green	none	motor supply OK
HV	orange	none	motor supply under voltage (<16V)
HV	red	none	motor supply over voltage (>98V)
HV	off	-	no motor supply
ST	green	none	drive OK and motor is energised
ST	orange	none	drive OK but motor is de-energised
ST	red	none	drive fault (see DF report for more information)
ST	red/green	1 second	
ST	red/green	0.25 second	communications receive buffer over-flow. Only a power-cycle clears this condition.
ST	red/off	0.5 second	only on power-up following a flash upgrade. Indicates bad truncated FPGA file.
ST	red/off	1 second	only on power-up following a flash upgrade. Indicates a CRC error.
ST	off	-	no logic supply. All other LEDs will also be off.
FB	green	none	fieldbus communications OK. Operational state.
FB	green	1 second	fieldbus communications OK. Pre-operational state.
FB	red	none	fieldbus communications fault
FB	off	-	no fieldbus option

Table 7-1. Status Bits Description (continued)

Forcing a Hardware RFS

Pin 2 of serial communications D-type connector X3 is for use as a hardware method of forcing a return to factory settings. It may be used when it is not possible to perform an OFF or RFS command. Such a situation may be switching to RS485 mode but having no RS485 interface on the controlling PC, forcing an RFS command will allow you to return to RS232 operation.

To force a hardware RFS follow the procedure detailed below, note you will loose any program in memory and system variables will return to their default values:

1. Turn off drive HV and +24V.
2. Connect X3 pin 2 (MODE input) to X3 pin 3 (0V).
3. Turn drive +24V on (and HV if required). On power-up, any program present in memory is cleared and ALL system variables are set to their initial factory default value.
4. Establish RS232 communications using Easi-V and type in 1SV and press [Enter] to store the changes.

Drive Faults

The following notes give you a better understanding of what is happening within the drive when a particular drive fault is reported. The explanations assume a ViX drive indexer firmware revision of V2.0.

Composite fault

This flag indicates that a drive fault has occurred and that the fault is still present. The original fault may have been registered by the FPGA (power card hardware) or by the drive's microprocessor. See the remaining fault status bits for the source(s) of the fault.

+/-15V Supply rail failure

This fault is detected by the drive hardware, which de-energises the power stage at the same time as informing the microprocessor of the fault.

Indexed motion will be stopped instantly and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.
The firmware will not allow an energise whilst this fault is present.

Motor HV under-voltage & over-voltage

The software monitors the HV every 500µS and compares the value to the under- and over-voltage trip values. If the HV is >98V or <16V, the following actions will be taken:
Indexed motion will be stopped instantly and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

The drive HV supply is measured when the MOTOR command is executed. If no HV is present (reading of 0 volts) then 80 volts is assumed to be the operating level. The reset threshold is then calculated as follows:

```
IF HV > 30 Volts THEN                 // if supply is normally greater than 30 volts...
reset_threshold = 30 volts            // HV must be >30 volts for ON command to work
ELSE IF HV >= 24 volts THEN           //
reset_threshold = HV * 0.85           // HV must be greater than (current supply voltage –
                                      // 15%) for ON command to be successful
ELSE IF HV >= 20.4 THEN               // supply is below nominal min but within tolerance...
      reset_threshold = 20.4 volts    // HV must be greater than absolute minimum
                                      // specification (24Vdc -15%) for ON to work
      ELSE
      reset_threshold = 16 volts      // set to trip out threshold
      ENDIF
```

This means that a drive fed from a > 30 volts HV supply will not be able to be energised via. the brake supply. Drives fed from a 24 Vdc HV supply need to have the brake supplied from a separate non-switched feed.

Vio over-voltage

Over-voltage will mean that the 24V supply is out of tolerance.
Indexed motion will be stopped instantly and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

Encoder/Auxiliary 5V under voltage

This 5V supply is read every 500uS. This fault is set if it dips below 4.5V.
Indexed motion will be stopped instantly and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.
The firmware will not allow an energise whilst this fault is present.

Impending Power loss (24v – Logic supply)

This input is triggered when the 24V supply is removed. The indexer will shut down when the 24V input drops below approximately 17.5V and will require a power cycle to recover.
At the moment that the impending power loss is detected, the following actions are taken:
Turn off interrupts (Drive will no longer communicate)
Turns off the power stage
Indicate indexer fault (STATUS LED = RED)
Loop forever until logic supply can no longer keep the micro alive.

If the logic supply is experiencing dips, then this may manifest itself as locked up comms. A common mistake when running off 24V HV, is to wire HV and logic to the same power supply. If an end stop is driven into, then the current drawn may be sufficient to collapse the HV and hence the logic supply. The drive faults with impending power loss, and de-energises the power stage, so the load is removed and the voltage recovers. The indexer however, appears to have locked up but has actually gone into a sleep mode.

Motor Over Temperature

Indexed motion will be stopped in a controlled manner and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

Ambient over temperature

When the temperature trip point is reached :
Indexed motion will be stopped in a controlled manner and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

Drive over temperature

When the temperature trip point is reached :
Indexed motion will be stopped in a controlled manner and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

Incompatible firmware revision

The FPGA firmware code contained in FLASH memory is not compatible with the controller hardware and the drive cannot be used.
This message is likely if in future, customers update old hardware with the latest FPGA firmware, which may require specific hardware to function.
The firmware will not allow an energise whilst this fault is present.

Unrecognised power stage

The power stage fitted to the drive is not recognised by this revision of firmware.
In this case, a customer may have down-graded their firmware. The required action is to update the drive firmware to a version that supports the fitted power stage.
The firmware will not allow an energise whilst this fault is present.

Controller diagnostic failure

This is set if the controller fails one of its self test routines. Further diagnostic information is available from the test commands.
The firmware will not allow an energise whilst this fault is present.

Output stage over current

This is monitored in hardware. The power stage will be de-energised by hardware and report the fault condition to the drive firmware.
Indexed motion will be stopped instantly and the drive firmware will then set the de-energise state in order to match the condition of the hardware.
If there is a user program running and the fault label is armed it will be run.
What might cause this fault?
The most common cause is poor or incorrect motor wiring. Other possibilities are a damaged motor winding or damaged power stage.

Output driver over current

The output stages are monitored in hardware and this bit will indicate that a fault has occurred (e.g. the output has been short-circuited).
Indexed motion will be stopped instantly and the drive will then de-energise.
If there is a user program running and the fault label is armed it will be run.

Tracking limit exceeded (Position Error)

For the ViXIM this drive fault indicates a stall condition, assuming stall detection is enabled.

A note about “Controlled stop”.
A controlled stop will be attempted if the trajectory generator was commanding motion at the time of the fault. If the commanded motion was due to following, then the indexer will disable following, that is there will be no ramp down of velocity.
If the drive was following and doing a superimposed move, then the superimposed move will be subject to the ramp down of velocity and once that has stopped, following will be disabled.

On base drives there is never any controlled deceleration of the motor. A fault will de-energise the drive.

On a ViXIM in MB mode, the trajectory generator is used to control the velocity, so those faults that lead to a controlled stop will act in a similar way to other indexed moves.

Returning the System

If a drive module is found to be faulty, you should contact your Parker Automation Technology Centre or the machinery manufacturer who supplied the product. Equipment for repair should NOT be returned directly to Parker without prior authorisation. Repairs will be carried out by Parker but will be processed via your supplier.

Parker may at their discretion authorise direct shipment to and from Poole, Offenburg or Rohnert Park, but only by prior arrangement with your supplier. Existing UK, European and USA customers who purchase equipment directly from Parker should contact Poole, Offenburg or Rohnert Park for further information (contact numbers are at the front of this User Guide).

8. Hardware Reference

Drive Specification – ViX250IM, ViX500IM

Functional Specification

Parameter	Value
Amplifier type	MOSFET chopper
User resolution	Freely programmable between 200 and 51,200 steps/rev
Nominal chopping frequency	16kHz
Protection circuits	Short circuit (phase-to-phase, across phases and phase to ground), motor overcurrent, over/under voltage, logic supply fault, over temperature, ext. 24V reversed, encoder fault
Maximum output current	ViX500: 5.6A rms (8A peak) ViX250: 2.8A rms (4A peak)
Output current adjustment	Programmable from 0.1A to maximum rated current
Standby current reduction	10% to 100% of programmed current (software-selectable)
Motor HV supply input	ViX500: 48-80V DC +5% -15% ViX250: 24-80V DC +5% -15%
Controller supply input	24V DC, 250mA (no outputs loaded, no encoder fitted). Fieldbus expansion module requires additional 50mA
LED status indicators (tri-colour)	HV fault, drive fault and comms status
Motor inductance range	0.5 - 10mH recommended

Table 8-1. Functional Specification

Indexer Specification

Parameter	Value
Command Interface	
Position range	+/- 2,147,483,647 steps
Velocity range	0.001 to 50 revs/sec
Acceleration/deceleration range	0.1 to 99999.99 rev/sec^2
Positioning modes	Incremental, absolute, registration, continuous run
Communication	
Data format	8 data bits, 1 start bit, 1 stop bit, no parity, optional echoback, Xon/Xoff supported
Baud rate	9600 / 19200
Address setting range	1 -255 by software
RS232 connection	2 wire plus ground
Digital Inputs	
User programmable inputs	5
Dedicated inputs	Home, + limit, - limit, registration, stop
Input levels (24V)	Logic high >14V, logic low < 4V
Input levels (5V)	Logic high >3.5V, logic low < 1V
Input impedance	4K7
Digital Outputs	
User-programmable outputs	3
Output levels output high output low	 +22V +10% -15% of supply 0.5V max (saturation of lower NPN transistor)
Output current rating	50mA maximum per output source 50mA per output sink

Table 8-2. Indexer Specification

Drive Environment Specification

Parameters	All drive types
Environment	Pollution degree 2, Installation category II
Operating temperature range	0 to 50°C ambient 0 to 40°C natural convection 40°C to 50°C airflow is required through the drive at better than 0.5m/s air velocity entering the drive.
Storage temperature range	-20 to 70°C
Humidity	5 to 95% non-condensing
Cooling	Natural convection
Housing	Plastic/Aluminium heatsink
Protection class	IP20
Weight	0.55kg

Table 8-3. Drive Environment Specification

Appendix A

Discrete Power Supply Recommendations

If the XL_PSU is not being used individual ViX drives can be powered from transformer/bridge rectifier power supplies of the type shown in Figure A-1. This design suggests suitable component values for powering particular drive types, but can be adapted to power more than one drive provided component power ratings are not exceeded.

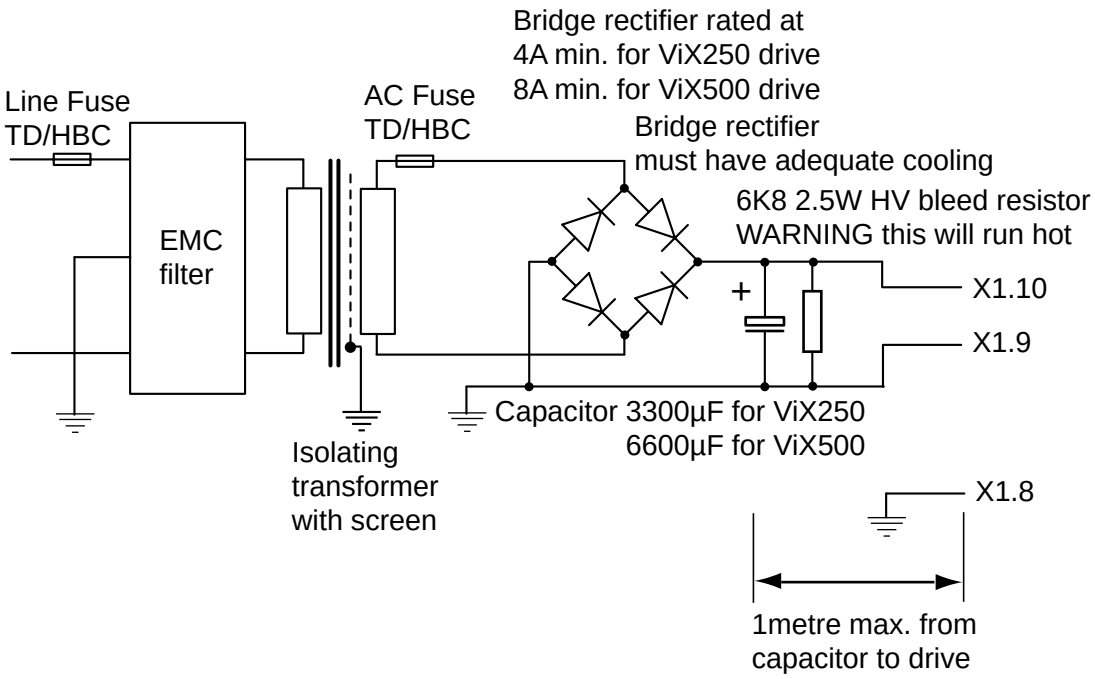


Figure A-1. Drive Power Supply

Note: The saftey earth must be reliably earthed at X1.8. The –DC input should also be connected to earth at a convenient point.

Supply Wiring

Use approved mains cable (at least 0.75mm^2) for primary wiring and route it away from secondary and signal wiring. Power input wiring must have a voltage withstanding rating of at least 1000V AC RMS. Note this figure is a test voltage, not the rated working voltage of the cable. Power input and motor wiring must be kept separate from signal wiring and insulated from operator access.

Note: secondary wiring must have a current rating in excess of the AC fuse rating.

Transformer Sizing for the DC Supply

A DC input is required by the drive, which can be generated by rectifying and smoothing the secondary voltage of a transformer. This is an unregulated supply so do not select a secondary voltage which generates a DC output greater than 80V.

The size of transformer required for a stepper drive installation depends very much on the application and on the maximum shaft power delivered by individual motors.

It is worth noting that in a one-off or low volume application it is usually preferable to be slightly generous in sizing the transformer, rather than spend a great deal of engineering effort trying to calculate the minimum possible rating. In low-power systems the potential savings in transformer cost are not large, although size and weight may also be a consideration. In a higher volume application the same principles apply to the prototype installation, but measurements of the supply current drawn under working conditions will give a useful guide to the final specification required.

Fuses

Fuses should be time delay, high breaking capacity and should be rated for the number of drives and expected in-rush current.

AC Fuse

The secondary AC fuse is intended to protect power supply wiring. Calculate the value of fuse required using:

$$\frac{\text{1.5 X secondary VA}}{\text{secondary voltage}} \quad \text{in amps}$$

Line Fuse

Fit line fuses to protect the transformer primary and associated wiring. If you cannot identify the live wire, fuse both phase conductors. Calculate the fuse value using:

$\frac{1.5 \times VA}{\text{supply volts}}$ in amps, but needs to be a minimum of 2A to cope with the in-rush current.

Fuse types should be anti-surge HBC (High Breaking Capacity).

Note: If the application requires a power dump, use a Digiplan power supply such as the PL1100.

CAUTION

Ensure that all power supply components are mounted away from operator access, as high voltages and hot surfaces are present in normal operation.

Index

S

T

Customer Feedback

If you have spotted any errors, omissions or inconsistent information within this user guide please let us know. Either use this page (or a photocopy) to describe the error and Fax. it to the number given below. Alternatively, you may phone or email the correction.

Name of user guide:

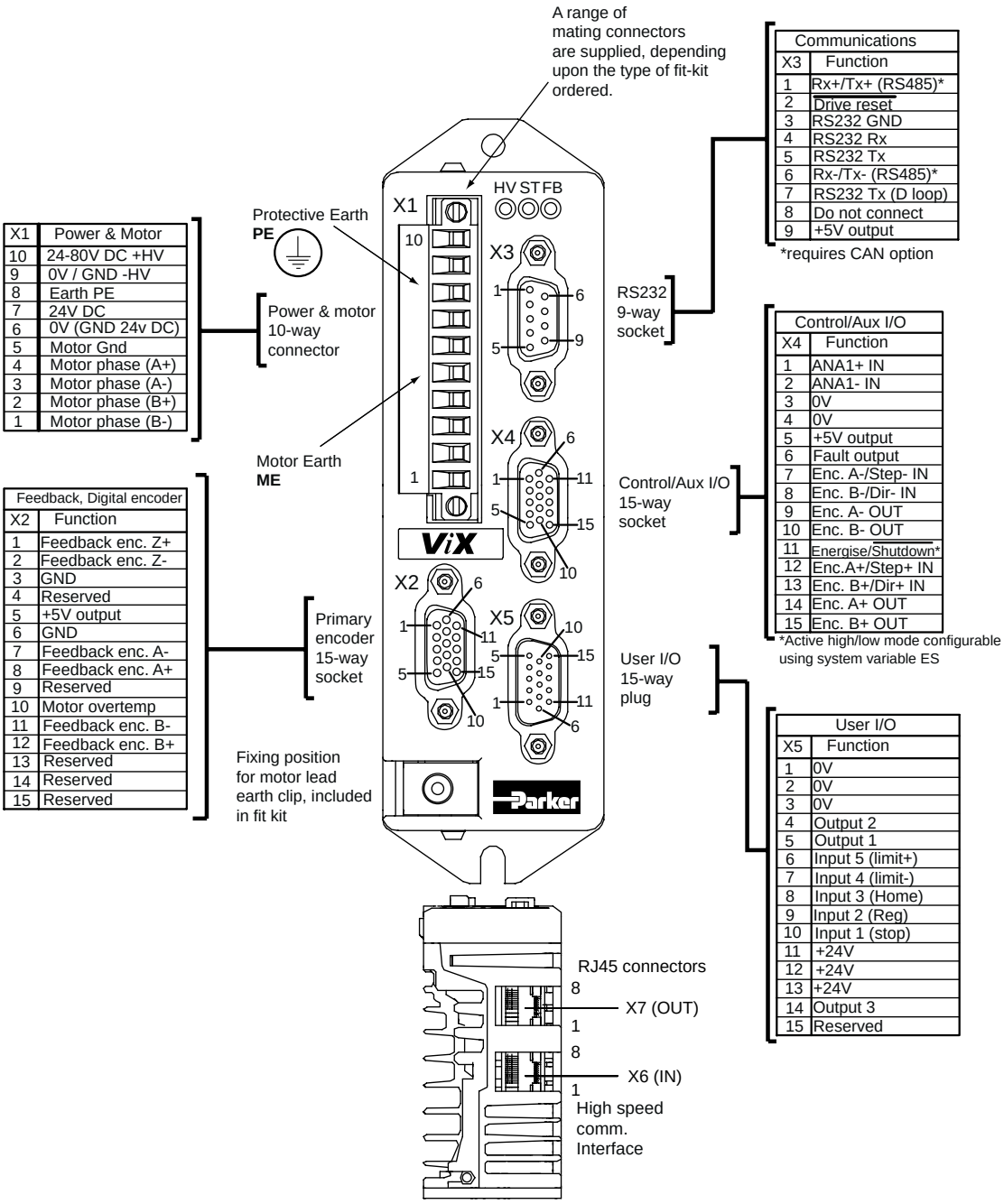
Part number: 1600. _ _ _ . _ _ Found on the title page in the bottom left corner.

Your name:

Contact number or email address:

Description of the error: (Please include page number)

Errors can be reported by Fax: **+44 (0)1202 695750**	By phone, via a technical support engineer: **+44 (0)1202 699000**	Or by email: **support.digiplan@parker.com**



X1	Power & Motor
10	24-80V DC +HV
9	0V / GND -HV
8	Earth PE
7	24V DC
6	0V (GND 24v DC)
5	Motor Gnd
4	Motor phase (A+)
3	Motor phase (A-)
2	Motor phase (B+)
1	Motor phase (B-)

Feedback, Digital encoder	
X2	Function
1	Feedback enc. Z+
2	Feedback enc. Z-
3	GND
4	Reserved
5	+5V output
6	GND
7	Feedback enc. A-
8	Feedback enc. A+
9	Reserved
10	Motor overtemp
11	Feedback enc. B-
12	Feedback enc. B+
13	Reserved
14	Reserved
15	Reserved

Communications	
X3	Function
1	Rx+/Tx+ (RS485)*
2	Drive reset
3	RS232 GND
4	RS232 Rx
5	RS232 Tx
6	Rx-/Tx- (RS485)*
7	RS232 Tx (D loop)
8	Do not connect
9	+5V output

*requires CAN option

Control/Aux I/O	
X4	Function
1	ANA1+ IN
2	ANA1- IN
3	0V
4	0V
5	+5V output
6	Fault output
7	Enc. A-/Step- IN
8	Enc. B-/Dir- IN
9	Enc. A- OUT
10	Enc. B- OUT
11	Energise/Shutdown*
12	Enc.A+/Step+ IN
13	Enc. B+/Dir+ IN
14	Enc. A+ OUT
15	Enc. B+ OUT

*Active high/low mode configurable using system variable ES

User I/O	
X5	Function
1	0V
2	0V
3	0V
4	Output 2
5	Output 1
6	Input 5 (limit+)
7	Input 4 (limit-)
8	Input 3 (Home)
9	Input 2 (Reg)
10	Input 1 (stop)
11	+24V
12	+24V
13	+24V
14	Output 3
15	Reserved