



CE

**XL25i/XL50i/XL80i
Stepper Drives
User Guide**

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.

The information in this user guide, including any apparatus, methods, techniques, and concepts described herein, are the proprietary property of Parker Electromechanical Division or its licensors, and may not be copied, disclosed, or used for any purpose not expressly authorised by the owner thereof.

Since Parker Electromechanical constantly strives to improve all of its products, we reserve the right to modify equipment and user guides without prior notice. No part of this user guide may be reproduced in any form without the prior consent of Parker Electromechanical Division.



Product Type: XL25i, XL50i & XL80i

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 XLi stepper 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.

WARNING – Risk of damage and/or personal injury

The XL drives and XL_PSU power supply 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.

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 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 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

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: tech_help@cmotor.com

Website: www.compumotor.com

Symbols used, have the following meanings:



Contents

1. Introduction.....	1
2. Mechanical Installation	3
3. Electrical Installation.....	9
4. Control of XLi Drives	41
5. EASI-Tools Software	75
6. Command Reference	89
7. XLi Maintenance and Troubleshooting.....	139
8. Hardware Reference	145
Index.....	151

User Guide Issue Change Summary

This user guide, version 1600.284.02, is the second version of the XL25i/XL50i/XL80i Stepper Drives User Guide.

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.

Major changes in revision 02 include:

Addition of XL80i information
SY motor range
XL_PSU information

Latest Changes Sheet

Known changes required to be made to this issue of user guide are listed below:

Note: This sheet is for documenting any changes or corrections necessary in this issue of user guide. Any necessary changes we become aware of will be incorporated into the main part of the user guide when it is next issued.

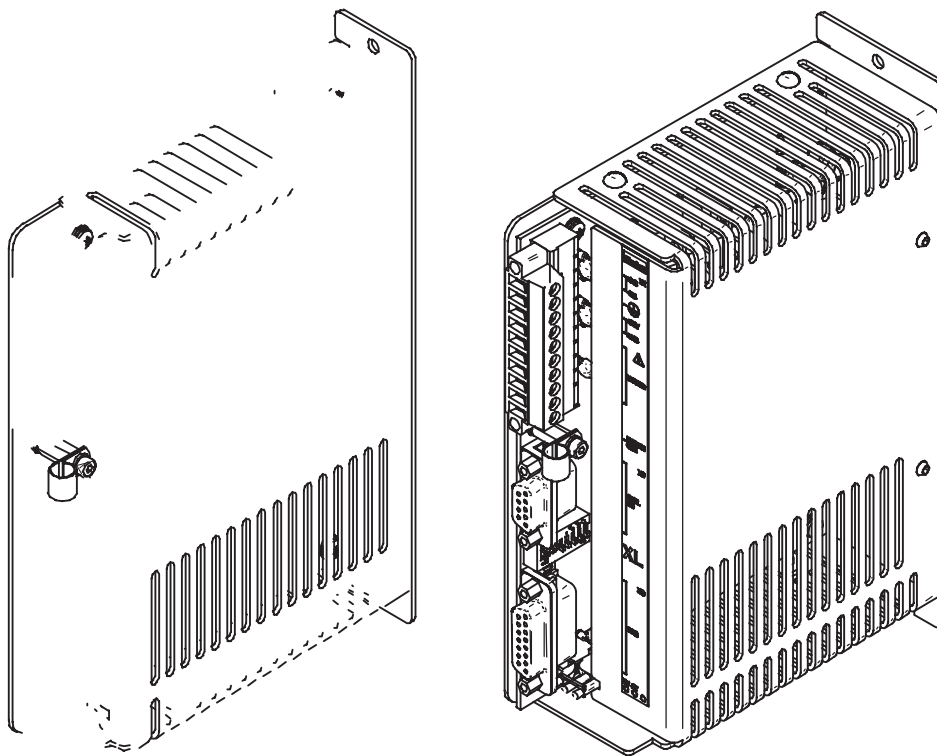
1. Introduction

XL25i/XL50i/XL80i Product Description

XL25i, XL50i and XL80i are intelligent low powered stepper drives offering a cost effective solution for applications involving single or multiple axis solutions. Designed to be used with existing general purpose stepper motors, the drives can be powered from a 20-84V (XL25i), 40-84V (XL50i) or 40-84V DC supply and are fully LVD compliant. EMC compliance is achieved when the drive is used with an external line filter and following the wiring and layout recommendations described in the section **Electrical Installation**. Three models provide an output current range of 1.25 to 8.0A:

- XL80i providing a motor voltage of 80V at 8A peak
- XL50i providing a motor voltage of 80V at 5A peak
- XL25i providing a motor voltage of 80V at 2.5A peak

The XL25i and XL50i share a similar case size, while the XL80i is slightly wider. Figure 1-1 illustrates the two case styles. See **Mechanical Installation** for dimension information.



An integrated indexer offers precise control of moves and sequences using a mix of parameter driven and event driven commands. A 24V DC I/O system provides high noise immunity, with 8 configurable inputs/outputs. Control is established using RS232, or via the Fieldbus.

The CAN variants are called:

XL80CAN

XL50CAN

XL25CAN

Software design enables application programs to be written with a minimal amount of user code. Pre-defining move profiles greatly reduces the amount of repetitive code that needs to be generated and the REG command allow registration moves to be accomplished in just a few lines of code. Event driven code also simplifies code generation by removing the need for long conditional testing sequences.

Product Features

Protection Circuits

- Motor short circuits
- Overvoltage
- Overtemperature
- 24V reverse supply protection
- Controller output short circuit

Function Indicators

- Drive Status (HV)
- Indexer Status (ST)
- Fieldbus Status (FB) - optional

Outputs and Inputs

- 8 digital inputs/outputs combined

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 Customer Specific Training Workshops, details of which can be found 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 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

XL25i/XL50i and XL80i low power drives have been designed for use in a temperature range of 0° to 50°C Max, 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.

It is recommended the drives are powered from the XL_PSU, which provides all the necessary power supplies and eliminates the need for an EMC filter. One XL_PSU can supply up to six drives alternatively, drives can be powered from a DC supply, fed from an isolated transformer. The insulation of the transformer must be suitably rated for the Installation Category and working voltage on the primary.

Drive Cooling

XL25i and XL50i drives are cooled by natural convection, whilst the XL80i is fitted with an integral fan. To aid cooling, drives should be installed vertically in an area where there is at least a 50mm air gap above and below the package and a 10mm gap either side. Avoid mounting heat producing equipment directly below a drive.

In the final installation, check that the ambient temperature specification of 50°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 a drive dissipation figure of 15W can be assumed for the XL25i/XL50i and 25W for the XL80i.

Drive Dimensions

XL25i and XL50i drives share the same dimensions, which are shown in Figure 2-1.

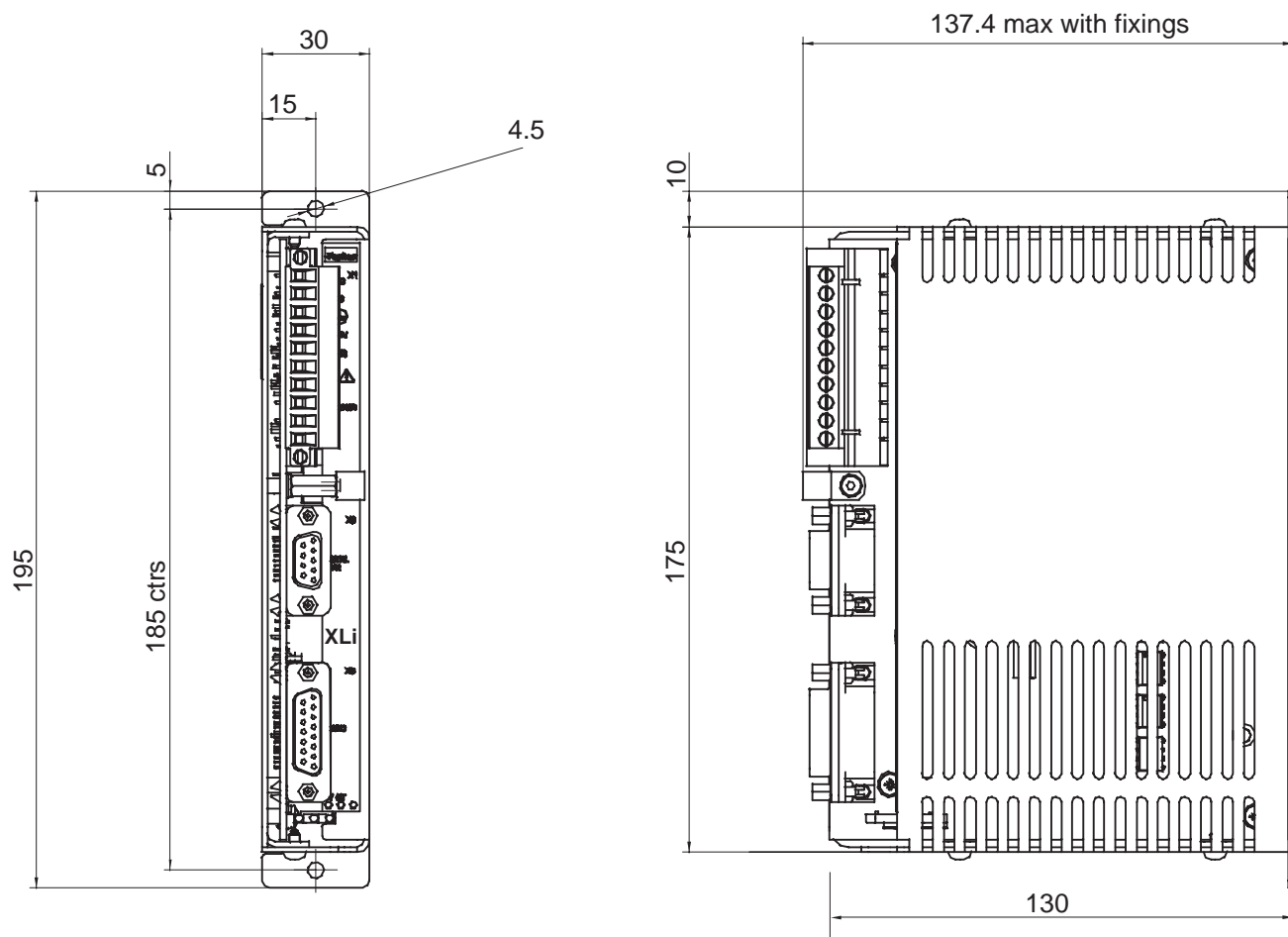


Figure 2-1. XL25i/XL50i Dimensions

Note: Later versions of this drive will have mounting slots at the base rather than holes.

The XL80i has the dimensions shown in Figure 2-2. Note the double slot fixing used on the base of this wider drive.

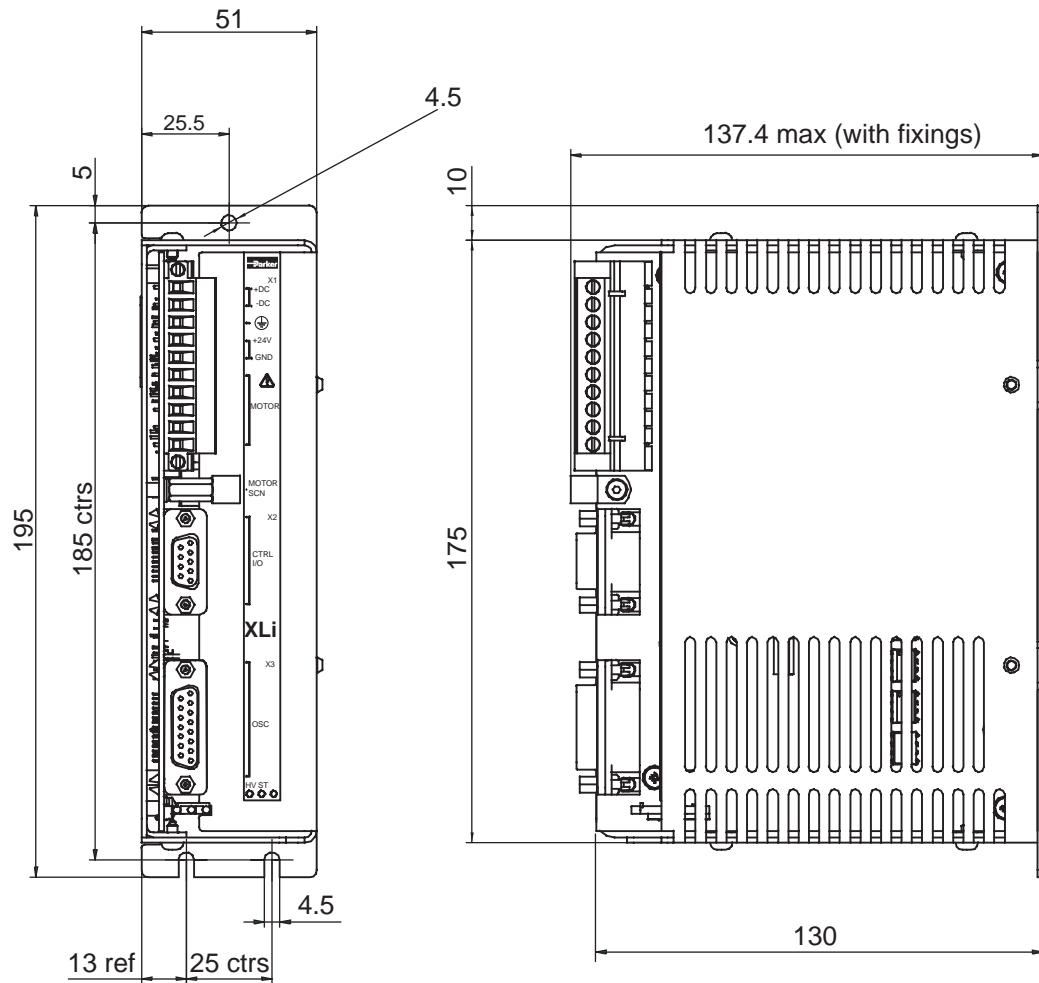
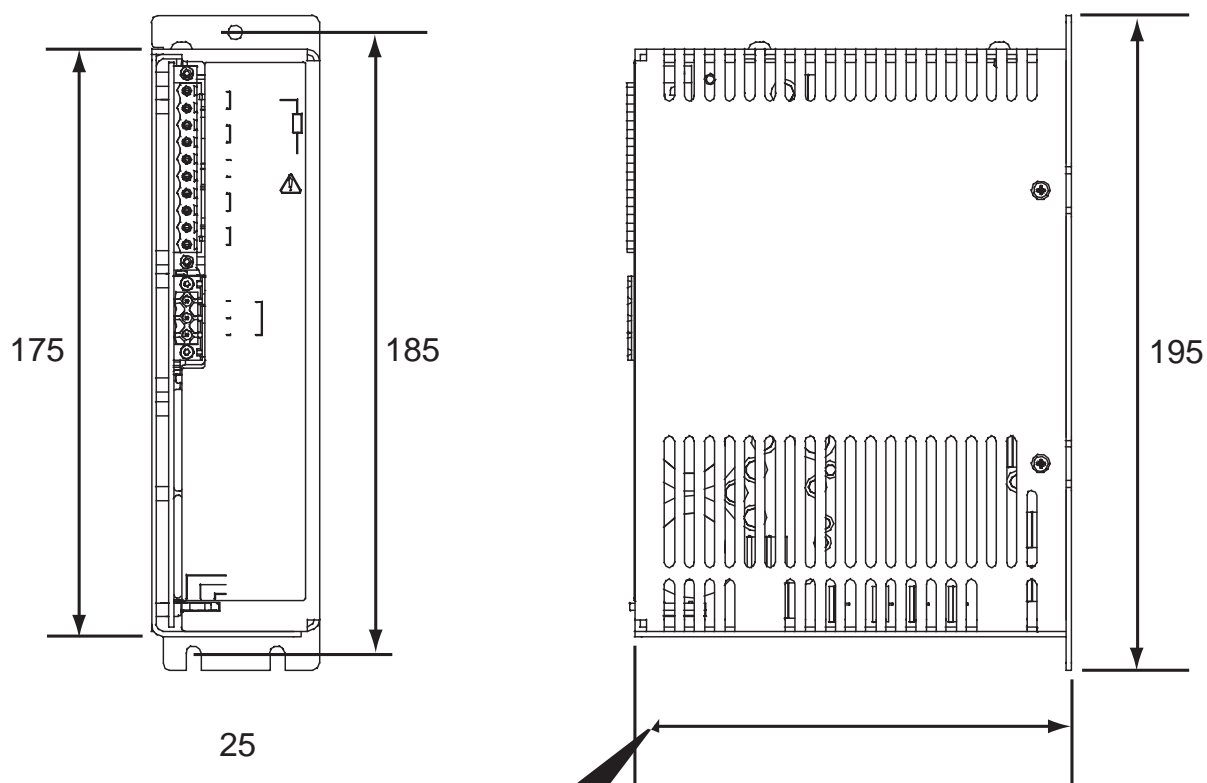


Figure 2-2. XL80i Dimensions

XL_PSU Dimensions

The XL_PSU dimensions are shown in Figure 2-3.



Motor Mounting Mechanical Considerations

Motors should be 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 may cause the motor shaft to be mechanically rotated in a manner uncontrolled by the drive.

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

Back driving the motor at low to medium speeds may establish sufficient voltage on the drive's capacitors to energise the drive.

3. Electrical Installation

Installation Safety Requirements

XLi stepper drives meet the requirements of both the EU Low Voltage Directive and EMC directive 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 motor and drive power when the cabinet door is open.

Precautions

During installation, take the normal precautions against damage caused by electrostatic discharges. Earthed wrist straps should always be worn.

User I/O connection 7 should be screened when used for registration.

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 it is recommended 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.

Individual drives are designed to be mounted on a metal earth plane to which the EMC filter's earth connection is also attached. 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.

Power Supply Connections

XLi drives must be supplied from a DC supply derived from an isolating transformer or a DC power supply.

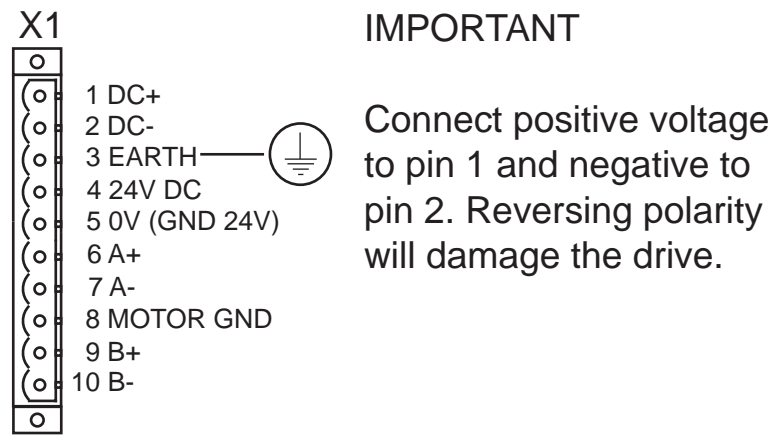


Figure 3-1. X1 Connections

Mating connector type is: Wieland 8213B/10 F OB, Part number 25.323.4053.0

Supply Requirements

XLi drives must be powered from DC supplies only as specified below:

Drive Type	DC Supply Voltage between DC+ and DC-
XL25i	20 to 84V absolute limits
XL50i	40 to 84V absolute limits
XL80i	40 to 84V absolute limits

Table 3-1. Drive Supply Voltages

Note: DC+ identifies the positive side of a DC supply.

The power supply must be isolated from the mains.

Safety Earth Requirements

The drive must be earthed, using the earth pin on X1 (pin 3).

XL Power Supply

The XL_PSU module is a 250W, power factor corrected, switched mode power supply for use with XL stepper drives. Designed for direct operation from world-wide single phase AC input voltages, the supply is capable of powering up to six drives without the need for an EMC mains input filter*. The use of the XL_PSU offers the following benefits:

- Auto-adapts to supplies between 95 and 264V AC
- No EMC filter*
- Less cabinet space
- Built-in +24V DC supply

*For drives with up to 30 metre motor leads.

Physical Appearance

Housed in a slim, 50mm-wide metal case, the XL_PSU is shown in Figure 3-2. This form of case matches the appearance of the XL drives and has the same height and depth.

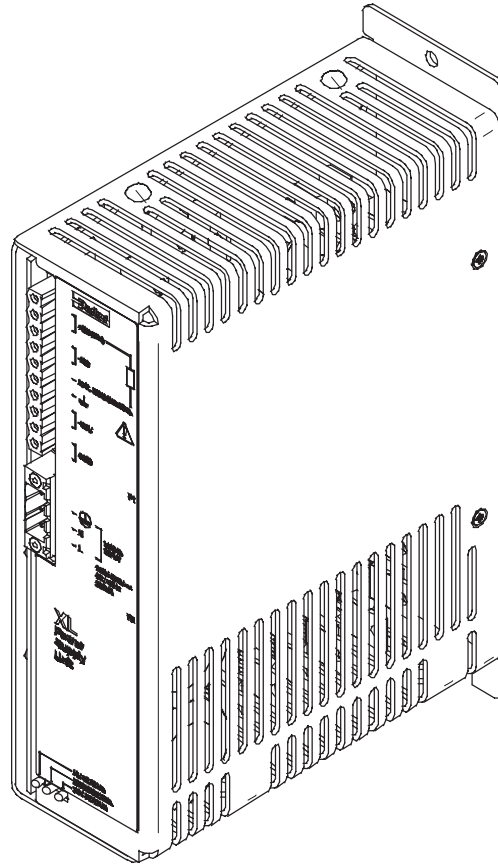


Figure 3-2. XL_PSU Switched Mode Power Supply

AC Supply Connections

Input AC supplies need to be connected to the 3-way mains input connector P2. Use approved mains cable, with a minimum wire size of 0.75mm^2 . The supplied mating connector is a Phoenix Contact, having the name MSTB 2,5/3-STF-5,08 and order number 1777992.

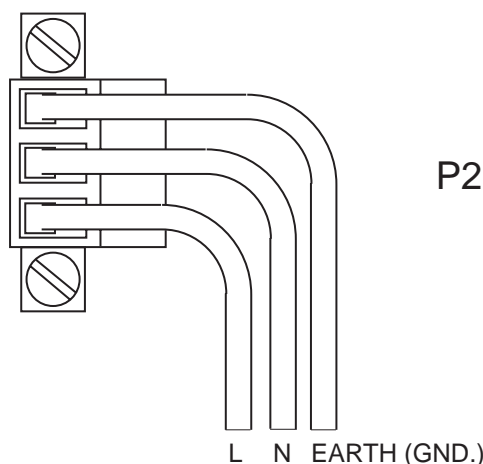


Figure 3-3. P2 Power In Socket Wiring

Output Connections

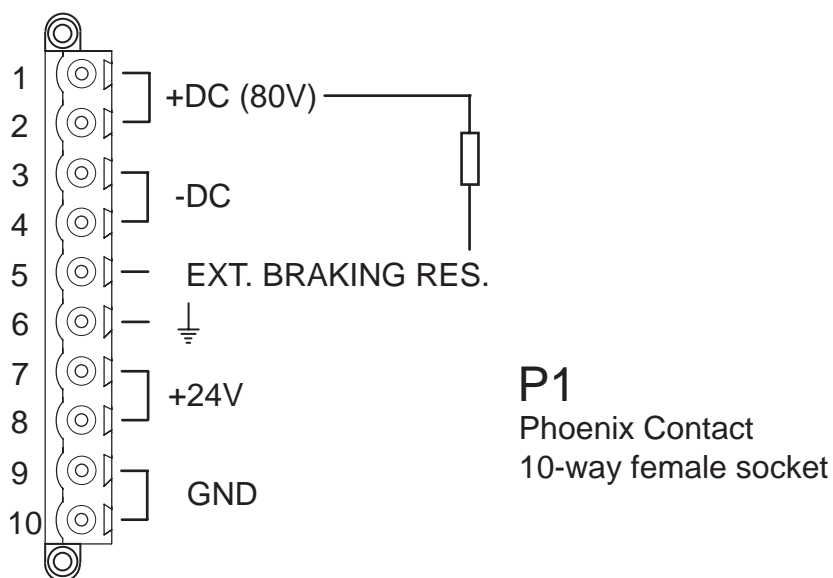


Figure 3-4. P1 Output Power Connector

Power out (+80V) is taken from pins 1 and 2 (use $32/02\ 1\ \text{mm}^2$ cable), the return connection being pins 3 and 4. +24V is available on pins 7 and 8 with the return connection being made to the GND pins 9 and 10. The earth pin 6 is internally connected to the -DC pins 3 and 4. The supplied mating connector is a Phoenix Contact, having the name IC 2,5/10-STF-5,08 and order number 1825394.

No connection exists between the earth pin 6 and the GND pins 9 and 10

Supply/Drive Connections

When used to supply up to six drives the power supply can be wired as shown in Figure 3-5.

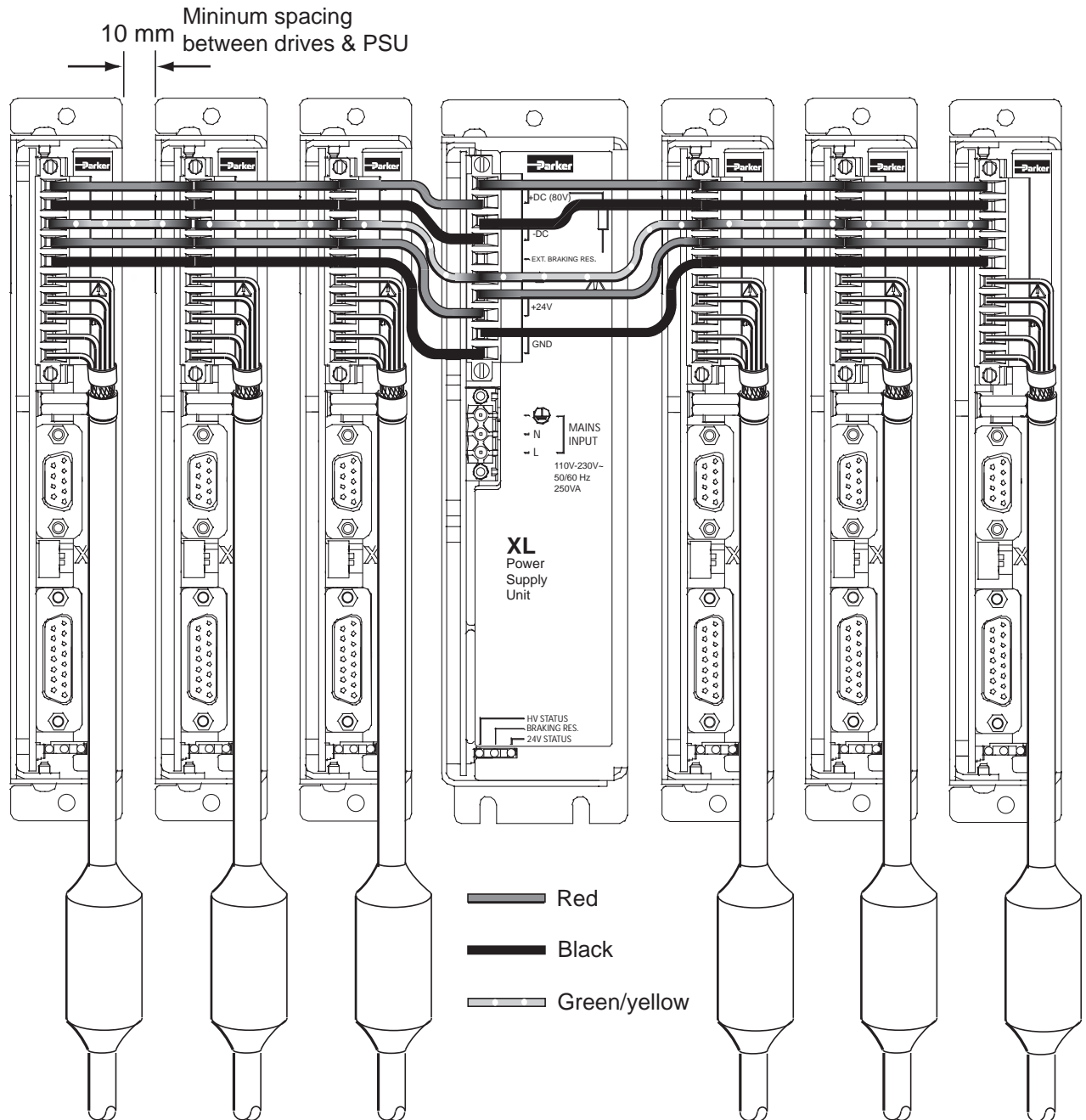


Figure 3-5. Power Supply and Drive Connections

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

Power Dump/Braking Resistor Considerations

The XL power supply incorporates a regenerative power dumping circuit which may be used to divert regenerated power into a dump or brake resistor. The need for such a circuit should ideally be determined during system design, but the calculations are repeated here for reference.

Calculating the Need for a Braking Resistor

For one drive axis a dump resistor will be required if the deceleration time in seconds is less than $(J\omega^2 - 0.2)$, where J is the total inertia in kg.m^2 and ω is the maximum speed in revs/sec. In a multi-axis system where a number of axes are decelerating at the same time, a given axis will not pump the HV up, provided that the deceleration time on that individual axis is greater than $J_1\omega_1^2$, where J_1 is the inertial of axis 1 and ω_1 is the angular speed of axis 1. Generally, other operating axes will absorb power from the supply and will add greater capacitive loading and will therefore reduce the need for power dumping. If it is known that certain conditions will cause several axes to decelerate together (say axes 3 & 4), then check that the $J\omega^2$ value minus the deceleration time for each axis (both 3 & 4), added together, do not give a result that is greater than 0.2. If they do, a dump resistor is likely to be required.

External Braking Resistor Installation

If required, an external braking resistor may be fitted between pins 1 and 2 of P1 and pin 5 of P1, as shown. The need for a braking resistor should be considered if the application requires large inertial loads to be decelerated quickly. During deceleration if the drive faults-out with over voltage or the braking resistor LED flashes orange - requesting power dumping, it is likely a braking resistor needs to be fitted.

Note: If a braking resistor is fitted, a ferrite ring-core needs to be fitted over the braking resistor leads, as shown in Figure 3-6. A suitable ring-core is the Parker Chomerics part number H8FE-1115-NC, as used for the drive's motor cable. The ferrite may be held in place using heat shrinkable tubing, or cable ties. An in-line 5 X 20 mm fuse is recommended to be fitted between pin 2 of P1 and the braking resistor. The fuse should have a maximum value of 3.15A with a time delay low breaking characteristic.

The braking resistor should be wired using 16/02 0.5 mm² cable.

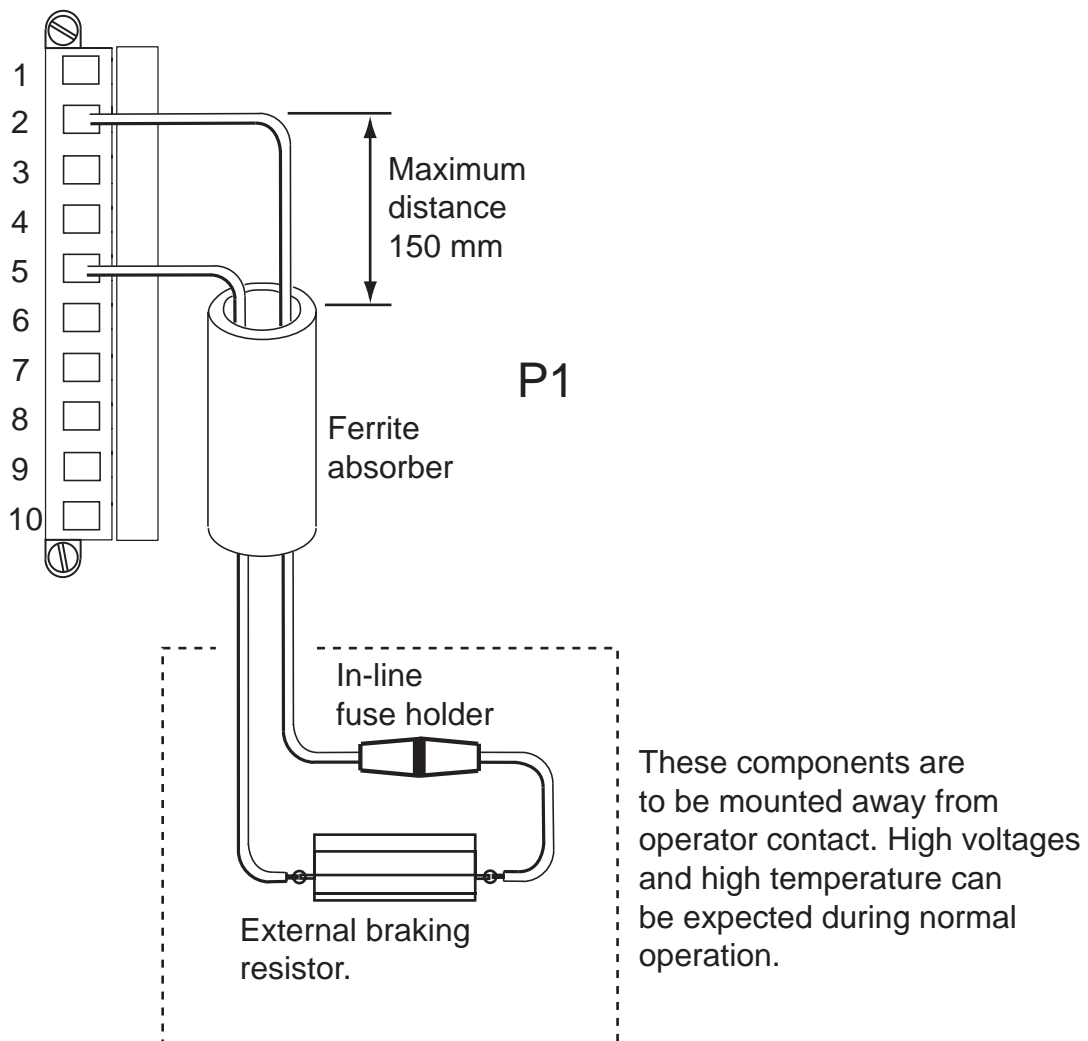


Figure 3-6. Braking Resistor Wiring

An internal dump switch has a peak rating of 800W. The dump resistor used should be 10 Ohms (nominal) with a power rating of 100W.

LED Indicators Function

Three tri-state LEDs are positioned on the front panel of the XL_PSU. Their function is described in Table 3-2.

Position	Colour	Function
Left	Orange	AC Input present and PSU fault. No HV output
	Green	AC Input present
Middle	Orange	Dump request
	Green	80V DC present
Right	Red	24V disabled by over current fault
	Orange	24V supplied by external supply
	Green	24V supplied by XL_PSU
	No colour	No 24V present

Table 3-2. LED Indicators Function

WARNING – Risk of damage and/or personal injury

The XL drives and XL_PSU power supply 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.

Discrete Power Supply Recommendations

If the XL_PSU is not being used individual XLi drives can be powered from transformer/bridge rectifier power supplies of the type shown in Figure 3-7. 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.

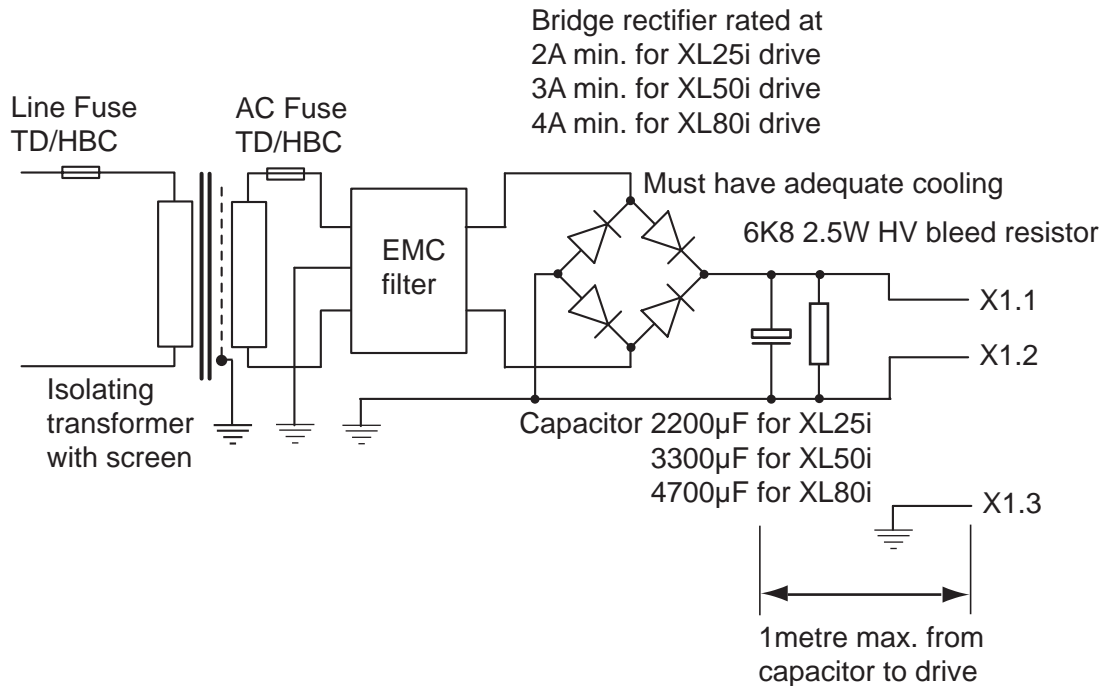


Figure 3-7. Drive Power Supply

Note: The -DC input must be reliably earthed at X1.3.

The EMC filter may be placed on the primary side of the transformer, assuming the voltage and current rating of the filter allow this.

Supply Wiring

Use approved mains cable (at least 0.75mm²) 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. However, the following figures may be taken as a guide.

Assuming a torque margin of 25-30% has been allowed, the maximum demand from a single XL25i drive is approximately 80VA, 125VA for an XL50i and 170VA for an XL80i. These are peak figures, which in most practical situations, and particularly in point-to-point positioning systems, will cause the average power demand to be much lower than this. The peak power demand may be significantly higher if a high performance enhanced-hybrid motor is used.

In multi-axis applications, the peak demand will depend on the maximum number of axes running simultaneously; very little power is used by a stationary axis. You should allow approximately 10VA for a stationary XL25i axis, 15VA for a stationary XL50i axis and 25VA for a stationary XL80i axis. For example, a typical system with four XL50i drives in which no more than two axes are in motion at any one time would comfortably operate from a 250-300VA transformer.

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.

Low Voltage Operation

The drive specification allows for 24V DC -15%, which works out at a little over 20V. This is the lowest that the supply should ever fall to if the risk of an under voltage fault is to be avoided. Supply ripple could use up part or all of this tolerance, so make sure that any ripple on the DC supply does not cause the input to dip below 20V.

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 components and should have a value twice that of the bridge rectifier current rating with a minimum value of 5A to prevent nuisance blowing.

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:

$$\frac{15 \times VA}{\text{supply volts}}$$

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).

+24V Supply Connections

XLi drives require a logic supply of +24V DC at 800mA per drive. This supply is used to power the indexer 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 indexer active when no main supply is present.

The +24V supply is connected to X1.4 and the return to X1.5, the total wire length, from supply to drive, must not exceed 10m.

The +24V supply 0V line needs to be connected to system earth (0V) at some convenient point before the EMC filter input.

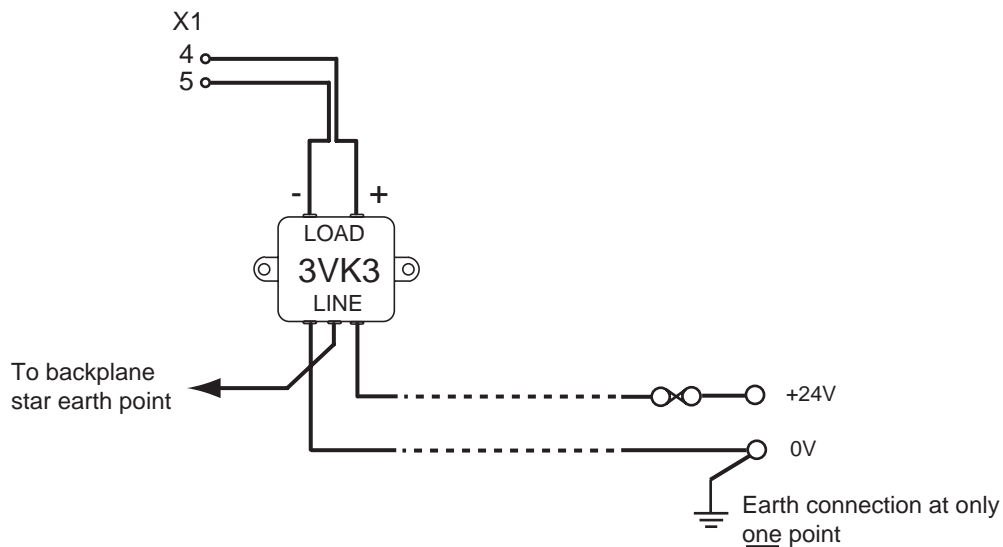


Figure 3-8. +24V Supply Connections

The supply should be fitted with a quick acting fuse, rated at 3A per single axis with a 1A increase for every extra axis up to a maximum of 16A. Note: The +24V supply used must meet the voltage requirements specification of +24V DC +10% -15%, ripple <1V p-p.

EMC Installation

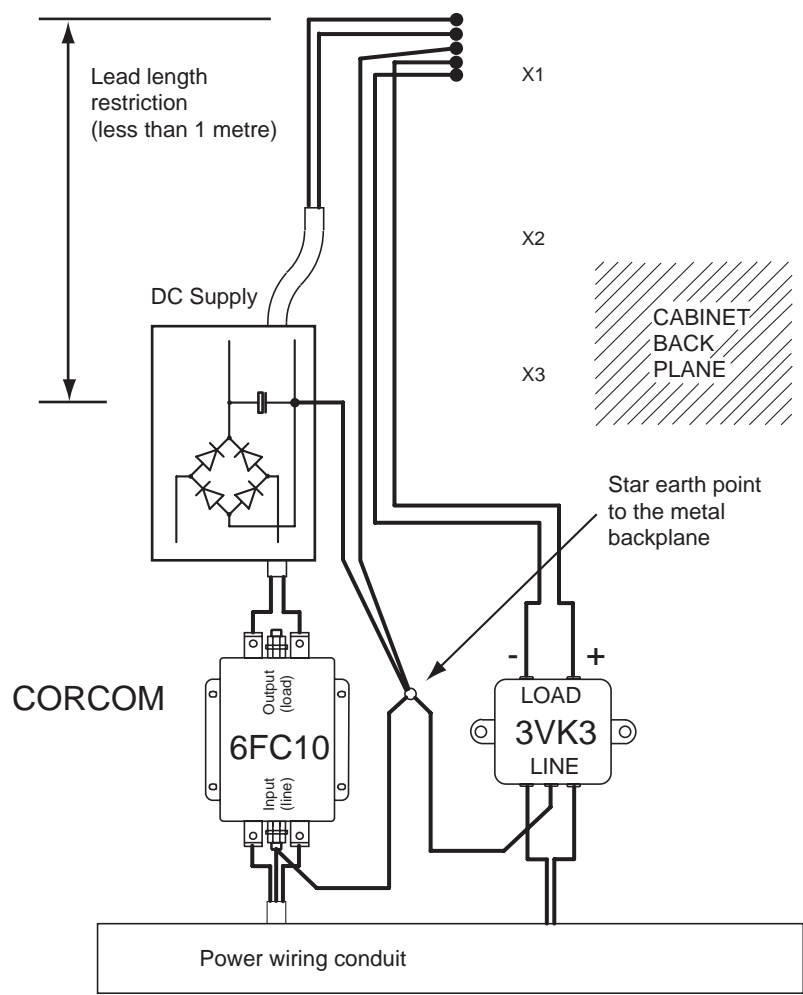
These EMC installation recommendations are based on the expertise acquired during the development of compliant applications, which Parker believes are typical of the way, a drive or drives may be used. Provided you have no special installation requirements or untypical operating environment requirements,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

General Requirements

XLi mounted stepper drives will require an EMC supply filter to meet EMC installation compliance requirements. It is recommended that the drive is mounted 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.

Filters should be mounted close to the drive and supply wiring kept 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. A typical EMC layout is given in Figure 3-9.

Recommended EMC filter types are CORCOM 6FC10 for loads up to 6A and 3VK3 for the +24V supply up to 3A. Larger systems may require higher current rated filters.



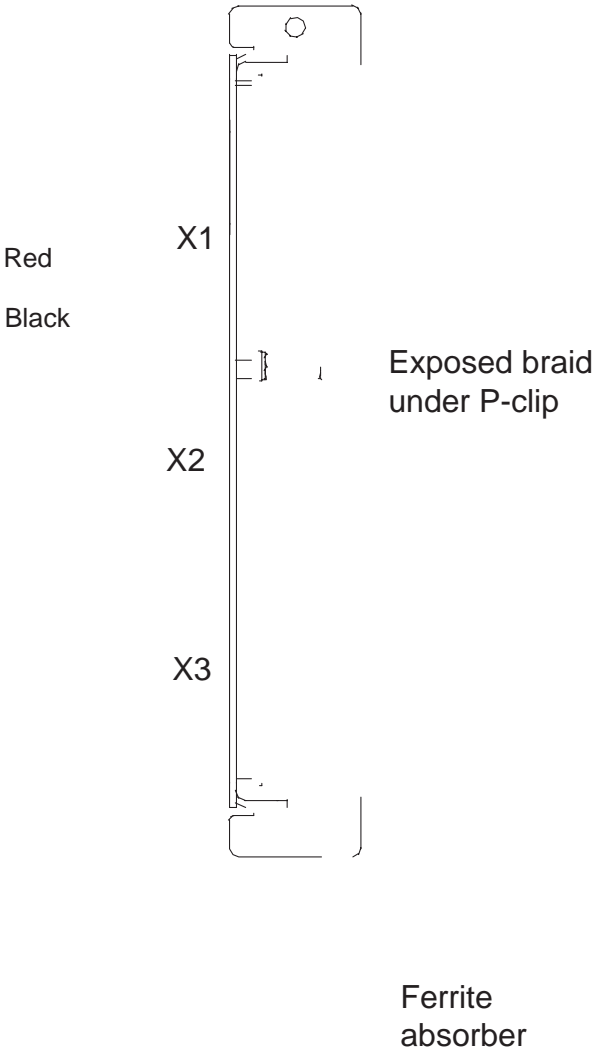
Motor Connections to the Drive

The recommended wire size for XL25i/XL50i/XL80i motor cables, of length less than 20m, is 1mm². For motor cable lengths greater than 20m (up to a maximum of 50m) use a wire size of 2.5mm². 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.

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.

At the drive end of the cable, a 360° connection to the screen should be made using the brass P-clip provided beneath the motor connector. The diameter of the P-clip is 6mm for use with the ready made motor cables having 1mm² conductors. 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 2.5mm² conductors the size of the brass 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 can be just below the drive, as shown in Figure 3-10. 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.**



Ferrite absorber specifications

The absorbers described in these installation instructions are made from a low-grade ferrite material which has high losses at radio frequencies. They therefore act like a high impedance in this waveband. The recommended component is produced by Parker Chomerics and is suitable for use with cable having an outside diameter up to 10mm. The specification is as follows:

Chomerics part number H8FE-1115-NC

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)

Control Signal Wiring

High-quality braided screen cable should be used for control connections. The cable screen of RS232 leads needs to be returned to the connector shell at both ends of the cable.

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, but remember the current rating of a motor is its DC rating, whilst drives are rated at peak output. To match a particular motor's current rating, multiply the drive's peak output current by 0.7. Drives can be derated to accommodate motors with lower current ratings (using system variable MC), 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 XL25i/XL50i/XL80i will drive motors having an inductance as low as 0.5mH, but the recommended motor inductance range is between 0.8mH and 10mH.

Performance of the XL25i/XL50i/XL80i is optimised for the following motor types, listed in Table 3-3 and Table 3-4.

Motor Type	Motor Rated Current in Amps*	Motor Inductance in mH per phase	XL80i	XL50i	XL25i
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	✓	✓	
SY1073	10.0	2.7	✓		

*(parallel connection)

Table 3-3. SY Optimum Motor Types

Motor Type	Motor Rated Current in Amps*	Motor Inductance in mH	XL80i	XL50i	XL25i
STT57-51P	3.0	2.0		✓	✓
STT57-83P	6.6	0.8		✓	
STT57-102P	6.5	1.0		✓	
STT83-62P	4.5	1.5	✓	✓	
STT83-93P	7.0	2.0	✓	✓	
STT83-135P	10.0	1.6	✓		
STT57-51M	1.4	9.5			✓
STT57-83M	2.7	5.7			✓
STT57-102M	4.1	2.5	✓	✓	
STT83-62M	2.4	5.2			✓
STT83-93M	5.6	3.2	✓	✓	
STT83-135M	5.4	5.6	✓	✓	

*(parallel connection)

Table 3-4. ST Optimum Motor Types

Motor Voltage Ratings

Motors with a withstand voltage rating from phase to earth of 1000V AC should be used. An insulated withstand rating of 500V AC is acceptable if an isolating transformer with earthed screen is used to power the system, and -DC 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 3-11. 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.

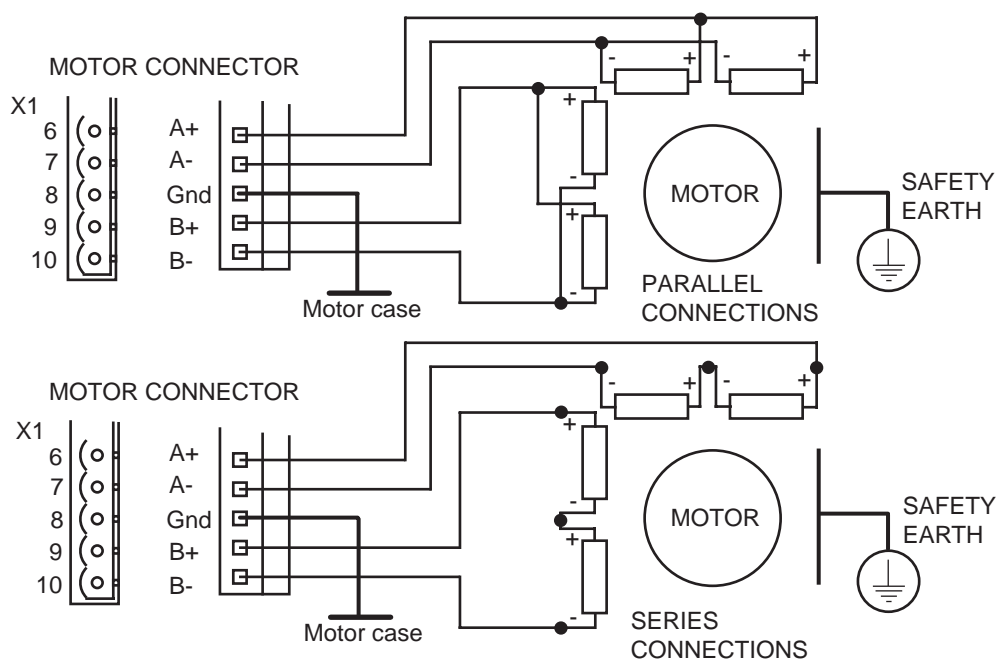


Figure 3-11. 8 Lead Motor Connection Options

WARNING - High Temperature

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

Motor Default Settings

Motor current is set by system variable MC, the default value being 50% of peak drive current. The resolution is set by MR and has a default value of 4000 steps/rev, whilst the standby current is set by MS and has a default setting of 50% of peak drive current. All default values are listed under the RFS command.

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 2.5mm² in area.

Motor Overload/Short Circuit Protection

The motor outputs are protected against overload and short circuits. The motor output will shut down within one second, when an overload of 250% of peak motor current is applied.

Note: In the case of stepper motors, locked rotor conditions do not cause an overload, so the only foreseeable failure mode is a short circuit which is immediately detected by the protection circuit.

Motor Wiring Identification Tables

The following tables supply serial and parallel connection information for a range of different motor types.

Note: The SY56 can be supplied as a 4-terminal motor, as shown in Figure 3-12.

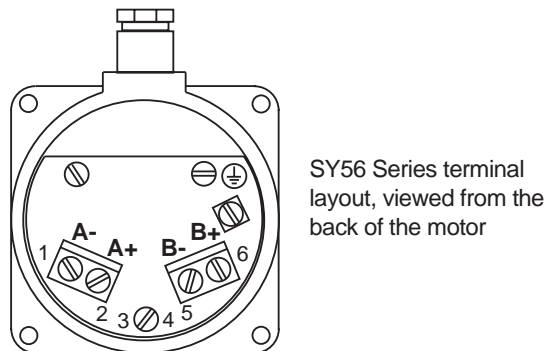


Figure 3-12. SY56 Motor Connections

MAKE	TYPE	A+	A-	B-	B+	NOTES
Parker EMD STT/STL	8-lead	Red	Yel	Pink	Blk	Link Blue & Violet, link White & Grey
	T.Box	1	2	3	4	Link 5 & 6, link 7 & 8
Parker OEM Series	4-lead	-	-	-	-	Internally wired in parallel
	8-lead	Red	Black	Green	White	Link Blue & Yellow Link Orange & Brown
Parker QM & S Motors	8-lead	Red	Black	White	Green	Link Yel & Blue Link Org & Brown
Parker QM & S Motors (except 106-205)	T.Box	1	3	4	5	Link 2 & 6 Link 7 & 8
Parker QM & 106-205 RS & TS	T.Box	1	3	7	8	Link 5 & 6 Link 2 & 4
Parker EMD SY 56 Series	8-lead	Blue	Green	Red	Brown	Link White to Black Link Grey to Yellow Internally wired in parallel
	T.Box	-	-	-	-	
Parker EMD SY 87/102 Series	8-lead	blue	green	red	brown	Link Yel & Gry, Link Blk & Wh
	T.Box	5	8	4	1	Link 7 & 6, Link 3 & 2
Pacific Scientific	6-lead	Black	Orange	Red	Yellow	White/Blk/Org,
	8-lead	Black	Orange	Red	Yellow	Link Wh/Blk & Wh/Org Link Wh/Red & Wh/Yel
	T.Box	1	3	2	4	Link 5 & 6, Link 7 & 8
Astrosyn, Rapidsyn, Slo-syn	6-lead	Red	Red/Wh	Grn	Grn/Wh	White & Black N/C
	T.Box (X6)	1	3	4	5	2 & 6 N/C
Slo-syn	8-lead	Red	Red/Wh	Grn	Grn/Wh	Link Black & White Link Org & Blk/Wh
	T.Box (X8)	1	3	5	4	Link 2 & 6, Link 7 & 8
Stebon	8-lead	Red	Yel	Pink	Blk	Link Blue & Violet, link White & Grey
	T.Box	1	2	3	4	Link 5 & 6, link 7 & 8
M.A.E	6-lead	Grn/W h	Grn	Red	Red/Wh	White & Black N/C
	8-lead	Black	Orange	Red	Yellow	Link Wh/Blk & Wh/Org Link Wh/Red & Wh/Yel
	T.Box	6	5	8	7	Link 1 & 3, Link 2 & 4
Zebotronics	T.Box	5	8	4	1	Link 2 & 3, Link 6 & 7
Oriental	6-lead	Black	Green	Red	Blue	Yellow & White N/C
Sonceboz	8-lead	Green	Grn/Wh	Red	Red/Wh	Link Org & Blk/Wh Link Black & White
Japan Servo	6-lead	Red	Blue	Green	Yellow	2 X White N/C
Escap	8-lead	Brown	Org/Wh	Red	Yel/Wh	Link Brn/Wh & Org Link Red/Wh & Yellow
Bodine	8-lead	Brown	Orange	Yellow	Red	Link Wh/Brn & Wh/Org Link Wh/Yel & Wh/Red
	T.Box	1	3	4	2	Link 5 & 7, Link 6 & 8

Table 3-5. Motor Connection Data - Windings in Series

MAKE	TYPE	A+	A-	B-	B+	NOTES
Parker EMD STT/STL	8-lead	Red & Blue	Yel & Violet	Wh & Pink	Blk & Grey	
	T.Box	1 & 6	2 & 5	3 & 8	4 & 7	
Parker OEM Series	4-lead	Red	Black	Green	White	
	8-lead	Red & Blue	Black & Yellow	Green & Org	White & Brown	
Parker QM & S Motors	8-lead	Red & Blue	Black & Yellow	White & Brn	Green & Orange	
Parker QM & S Motors (except 106-205)	T.Box	1 & 2	3 & 6	4 & 7	5 & 8	
Parker QM & 106-205 RS & TS	T.Box	1 & 5	3 & 6	2 & 7	4 & 8	
Parker EMD SY 56	8-lead T.Box	Blu & Gry 2	Grn & Yel 1	Red & Blk 5	Brn & Wh 6	
Parker EMD SY 87/102Series	8-lead T.Box	Blu & Gry 5 & 6	Grn & Yel 8 & 7	Red & Blk 4 & 3	Brn & Wh 1 & 2	
Pacific Scientific	6-lead	Black	Wh/Blk/ Orange	Red	Wh/Red/ Yellow	Org & Yellow N/C
	8-lead	Black & Wh/Or	Orange & Wh/Blk	Red/ Wh/Yel	Yellow & Wh/Red	
	T.Box	1 & 5	3 & 6	2 & 7	4 & 8	
Astrosyn, Rapidsyn, Slo-syn	6-lead	Red	Black	Green	White	Red/Wh & Grn/Wh N/C 3 & 5 N/C
	T.Box (X6)	1	6	4	2	
Slo-syn	8-lead	Red & White	Blk & Red/Wh	Grn & Blk/Wh	Org & Grn/Wh	
	T.Box (X8)	1 & 2	3 & 6	4 & 7	5 & 8	
Stebon	8-lead T.Box	Red & Blue 1 & 6	Yel & Violet 2 & 5	Wh & Pink 3 & 8	Blk & Grey 4 & 7	
M.A.E	6-lead	Grn/Wh	White	Red	Black	Grn & Red N/C
	8-lead	Black & Wh/Or	Or & Wh/Blk	Red & Wh/Yel	Black Yel & Wh/Red	
	T.Box	3 & 6	1 & 5	4 & 8	2 & 7	
Zebotronics	T.Box	5 & 6	8 & 7	4 & 3	1 & 2	
Oriental	6-lead	Black	Yellow	Red	White	Grn & Blue N/C
Sonceboz	8-lead	Grn & Blk/Wh	Or & Grn/Wh	Red & White	Blk & Red/Wh	
Japan Servo	6-lead	Red	White*	Green	White*	
Escap	8-lead	Brn & Orange	Brn/Wh & Org/Wh	Red & Yellow	Red/Wh & Yel/Wh	
Bodine	8-lead	Brn & Wh/Or	Wh/Brn & Or	Yel & Wh/Red	Wh/Yel & Red	
	T.Box	1 & 7	3 & 5	4 & 6	2 & 8	

*use correct white for each phase

For 6-lead motors, connections shown are for one half-winding.

N/C means No Connection

Table 3-6. Motor Connection Data - Windings in Parallel

Control and Communications Wiring

Figure 3-13 shows the XLi drive connector layout.

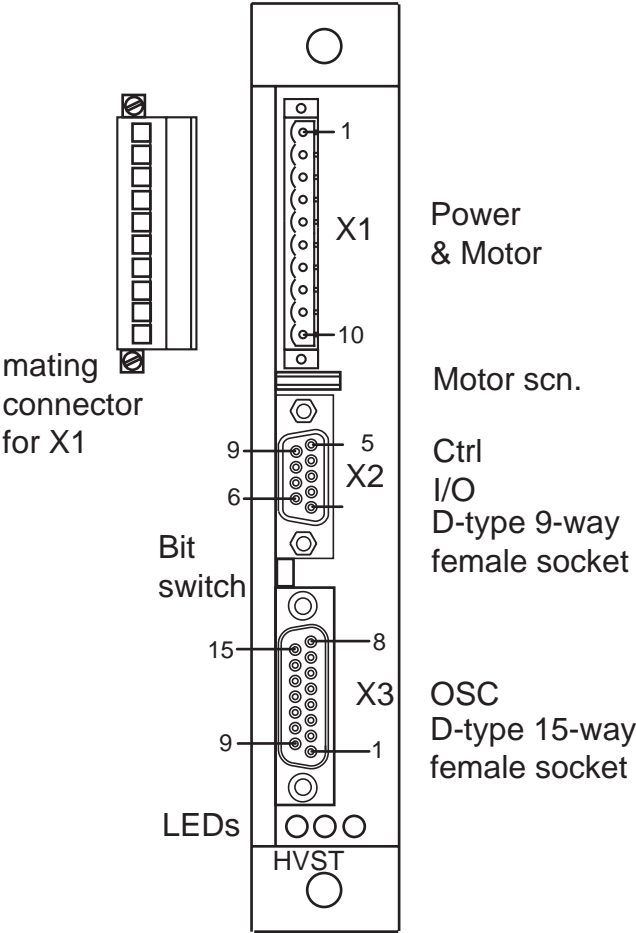


Figure 3-13. XLi Connector Pin Layout

Terminal Description

X3 Connector

User programmable inputs 1 to 8 are routed to the drive's indexer via connector X3. These connectors also provide access to the drive's externally connected 24V supply. Inputs and outputs share the same pin, that is output 1 drives input 1. If the output is OFF, the input can be used.

No screening of I/O wiring is necessary, apart from input 7 when used as a registration input. I/O lead lengths must not exceed 30m.

Connector Type

15-way D-type.

Connector Pin Out

Connector Pin	Signal Name
1	+24V
2	I/O 1
3	I/O 2
4	I/O 4
5	I/O 6 (Limit +)
6	I/O 8 (Home)
7	GND
8	GND
9	+24V
10	+24V
11	I/O 3
12	I/O 5 (Limit -)
13	I/O 7 (Registration)
14	GND
15	GND

Table 3-7. X3 Signal Connections

User inputs 5 to 8 (inclusive) can be used as general purpose user inputs or have a dedicated function. These inputs become dedicated for a particular use when their associated command is turned ON. The functions associated with each input are:

Negative limit	Input 5 (X3 pin 12)
Positive limit	Input 6 (X3 pin 5)
Registration	Input 7 (X3 pin 13)
Home switch	Input 8 (X3 pin 6)

For example, assuming axis 3 has already been sent a configure homing command, homing could be turned ON by issuing a **3HOME1** command. In this case, axis 3 input 8 becomes dedicated for use with the home switch on axis 3. This input cannot be used for any other purpose until the homing command is turned OFF and the input is re-defined (very unlikely if homing is being used). Inputs 5 and 6 automatically default to limit inputs (input 5 becomes LIM- and input 6 becomes LIM+), but they can be re-configured as general purpose user inputs by disabling both positive and negative limits in the LIMITS command.

Internally, inputs 5 and 6 are connected to virtual inputs 9 and 10. This allows a program to read limits from inputs 9 and 10 for upwards compatibility with DIN rail Li drives. Figure 3-14 shows the common connection of 8 inputs and outputs used by the XLi drive.

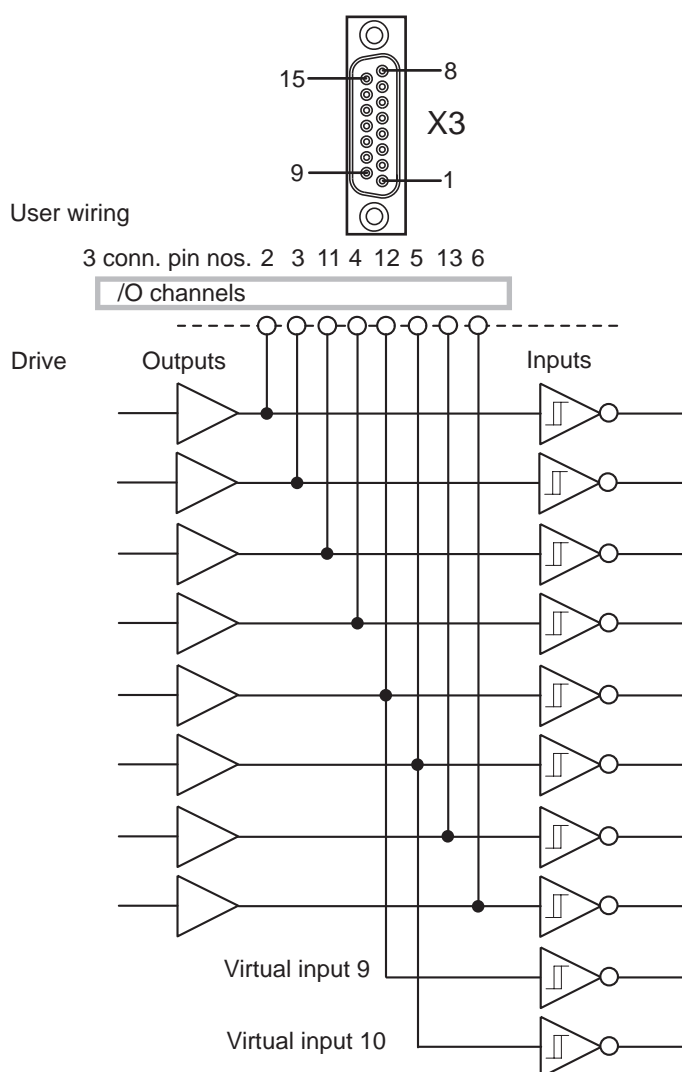


Figure 3-14. Input/Output Connections

Programmable Input Circuit

Eight high level logic inputs/outputs are provided for use with 24V PNP style control signals. Each individual I/O circuit has the input circuit arrangement shown in Figure 3-15. Switching levels are given in the indexer specifications.

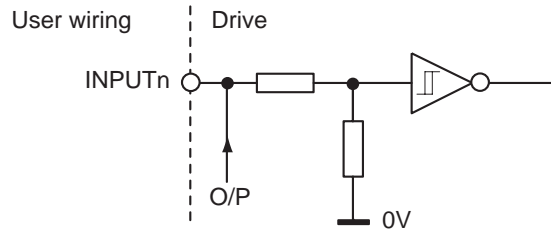


Figure 3-15. Programmable Input Circuit

This style of input can be used with PLC outputs or simple switch contacts as shown in Figure 3-16.

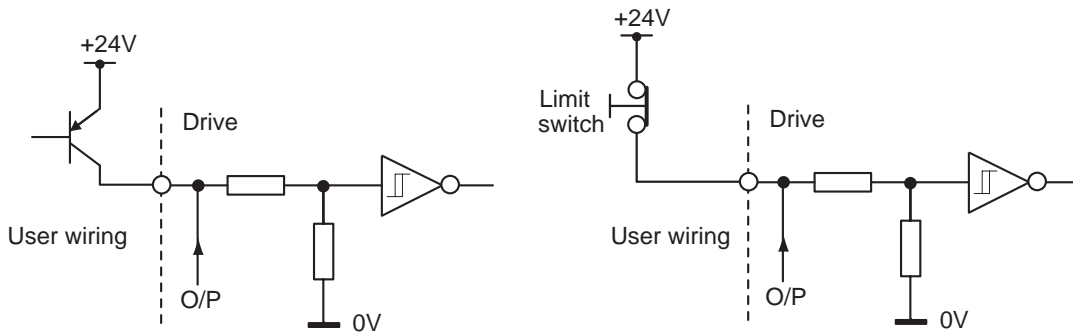


Figure 3-16. Input Driven from PNP Output or Limit Switch

Input Response Time

All inputs have a response time that is dependent upon the scan rate value programmed using the Scan Command (**SN**). The default value of this command is 50ms, which should be reduced if a faster signal input response is required.

User I/O Configuration

The XLi has eight logic input/outputs, one of which is dedicated for high speed use (REG). All user inputs and outputs operate with high voltage logic levels to maximise noise immunity.

User Outputs

Eight 24V user outputs are available shared with the inputs, having a circuit arrangement shown in Figure 3-17.

Do not set an output ON if the corresponding input is being used, that is, as a general input or for limits, home or registration.

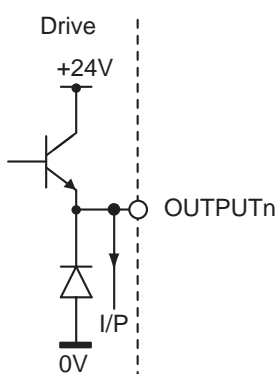


Figure 3-17. Output Logic Circuit

Outputs of this type can be used to drive relays directly, as shown in Figure 3-18. Each output is rated at a current of 60mA max, but this cannot be drawn by all eight outputs simultaneously. A current limit of 400mA applies to the total current drawn by all eight outputs operating together. A prolonged use of these outputs at current levels greater than 400mA in total, will cause the outputs to turn OFF (drop to 0V), a drive fault sequence will be run, if defined and programmed.

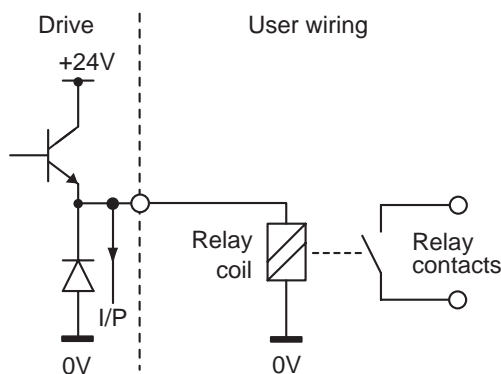


Figure 3-18. Output Logic Circuit

Note: the drive output circuit includes a protection diode to suppress relay coil back e.m.f effects. It is important that the 0V of the coil is referenced back to the I/O 0V connector.

Fanout Capability

Each output has a fanout capability of 3. Figure 3-19 shows one user output driving three user inputs, internal input and output decoupling capacitor values are also shown.

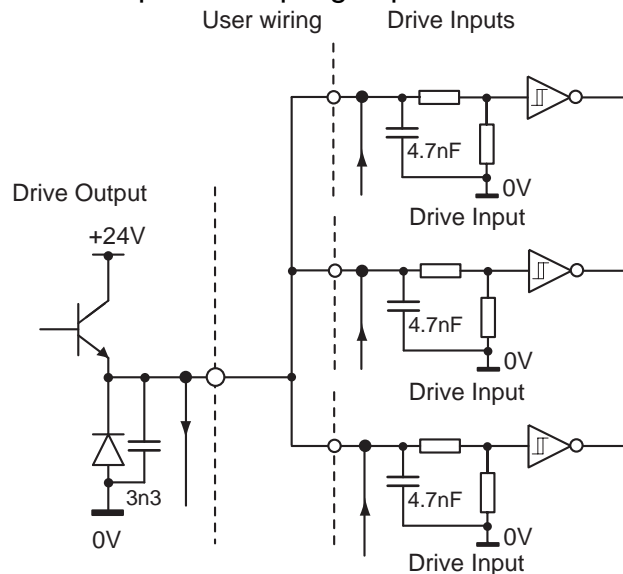


Figure 3-19. Maximum Output Fanout of Three

Fault Output

Any of the programmable outputs can be used as a Fault Output by setting a particular output to a 'no fault' condition at the beginning of a program. When a fault is encountered the fault routine can be used to change the state of the output to 'fault'.

For a complete explanation see the sub-section entitled **Fault Label** in **Section 4 - Control of XLi Stepper Drives**.

Motor Connector (part of X1)

Motor connections need to be made using a side entry plug, as specified below. For full motor cable wiring details see the EMC installation information.

Connector Type

Wieland 8213B/10 F OB, part number 25.323.4053.0

Connector Pin Out

X1 Pin Connection	Signal Name
6	Motor Phase A+
7	Motor Phase A-
8	Motor GND (drain)
9	Motor Phase B+
10	Motor Phase B-

Table 3-8. X1 Motor Connections

X2 Connections

X2, the Serial Communications Connector provides RS232 and CANBUS options. The pin out and signal names are shown in Figure 3-20.

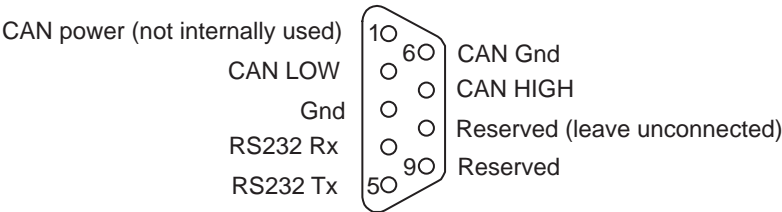


Figure 3-20. RS232/CAN Connector

All serial communication connections should be made using screened twisted pairs. The cable screen being returned to the connector shell at both ends of the cable.

X2 Connections (9-way D-type Male plug)

Pin	RS232/Optional CAN
1	CAN POWER
2	CAN-L
3	GND
4	RS232 Rx receive
5	RS232 Tx transmit
6	CAN GND
7	CAN-H
8	Reserved (leave unconnected)
9	Reserved

Table 3-9. RS232/CAN Connections

X2 PC/RS232 Connections

PC or Terminal RS232 connections are made to the X2 connector, using a screened lead wired as shown in Figure 3-21.

- Note:
- [1] Screen connections should be made to the connector shell at either end of the cable.
 - [2] X2 connectors are D-type 9-way female connectors.

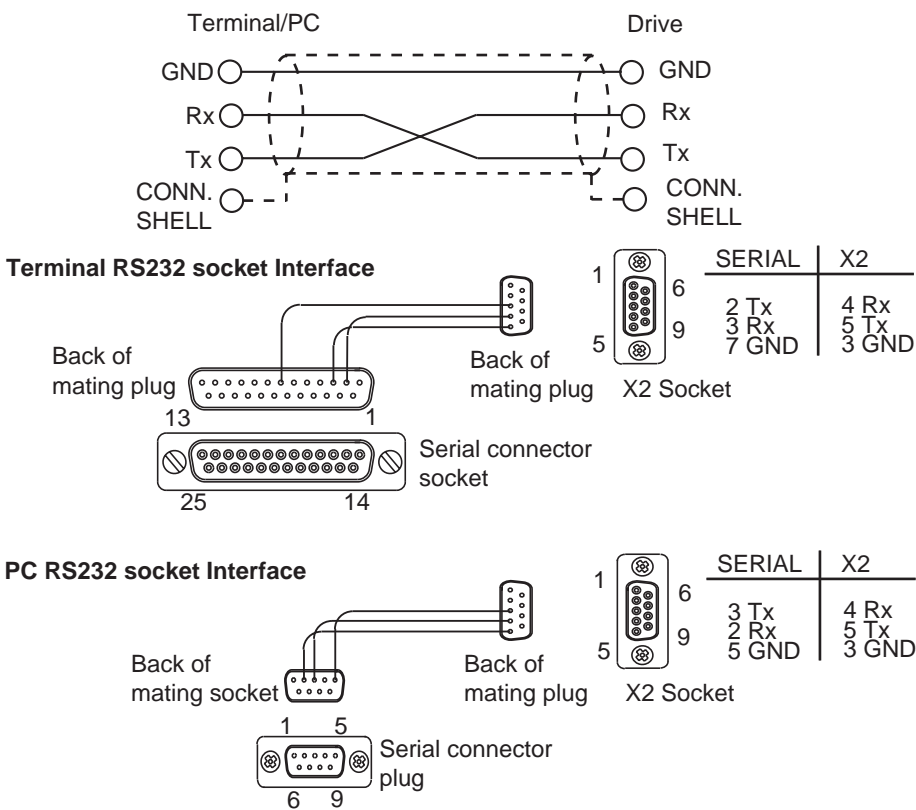


Figure 3-21. RS232 Lead Connections

Note: These serial connections do not follow standard RS232 pinouts. Do not attempt to use a null modem cable.

Drive Settings

Fixed at 9600 baud RS232. Software address selection, default address 1.

Software Address Selection

An RS232 address is set by software, using the # command. The address following the # command is saved in the drive's non volatile memory following an <a>SV command.

Software addressing is useful where a multi-axis system needs to be set-up. By issuing a #1 command the first indexer will send a #n+1 command along the daisy chain (assuming echo mode is set) resulting in each drive being given an address which increments around the daisy chain. Consequently, the first drive will be assigned address 1, the second address 2 and so on, until the final drive will echo back (final drive number +1). In this manner drives will be assigned addresses that increment in numeric value around the RS232 daisy chain connection.

New XLi drives will need their address to be configured individually, using the method described above. If they have been arranged in a daisy-chain, after power-on allow time for the RS232 bus to clear, and ensure the first command sent is #1.

To set addresses starting from a particular address, use the command #n, where n is the address you wish to start from. For example, to increment the daisy chain addresses from 6 onwards, use the command #6. If you wish to give a particular axis a new address, you can use the command <a>#n, where <a> represents the old address and 'n' is the new address.

When addresses are incremented using n+1, any previous, non-sequential, address pattern will be re-ordered as a sequential pattern. For example, if a system is composed of drives with addresses 3, 5, 7 (in that order) is sent the command #20, the addresses will be changed to 20, 21, 22. The maximum address that can be sent using incremental addressing is [255 - the number of axes being used].

4. Control of XLi Stepper Drives

Overview

This section introduces you to the operation of the XLi stepper indexer/drive, motion control concepts and the way certain software commands are used. Event driven code techniques are explained together with the use of system variables, also fault reporting and the interrogation of system variables are covered. Features of the drive are introduced in parallel with an explanation of move types and motion profiles. Registration operation, homing concepts, user I/O configuration are also covered, together with serial communications.

Compared with the DIN Rail Li drive, the XLi does not have BOOST and does not have encoder inputs, so no Position Maintenance, or Stall Detection operation is possible. The sharing of eight inputs and outputs does not allow thumb wheel commands and no RS485 or analogue inputs are available.

Controller Operation

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

Direct Mode

Direct control of the indexer 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 indexer 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.

Direct Mode control can be used when first testing a drive:

1ON	;enable the drive
1LIMITS(3,0,0)	;disable limits
1D4000	;set distance to 4000 steps
1G	;execute a pre-defined move

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. Alternatively, you could directly issue a **<a>GOTO(START)** command.

Code Structure

Program code is written 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 to be structured in the form illustrated in Figure 4-1, which shows the block nature together with an example of code.

Declare

Every label used in a program, apart from START, REG, NOREG and FAULT, needs to be declared before it can be used. 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.

Labels can only be declared 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. By declaring labels in the command line, before any START code, makes the most efficient use of the available memory, but if you then upload 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.

Example of DECLARE being used in the command line:

```
1K                ;Kill or stop any program currently running
1CLEAR(ALL)       ;Erase all 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 alpha characters terminated with a colon (:). 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).

Upto 14 labels can be used, although four of these have already been allocated to START, REG, NOREG and FAULT, leaving ten 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).

The LSEL command must be armed before input selection of labels will operate.

Structure

The code example of an absolute positioning move shown in Figure 4-1 demonstrates how code may be written that follows the block structure. Start code is first used to initialise the drive:

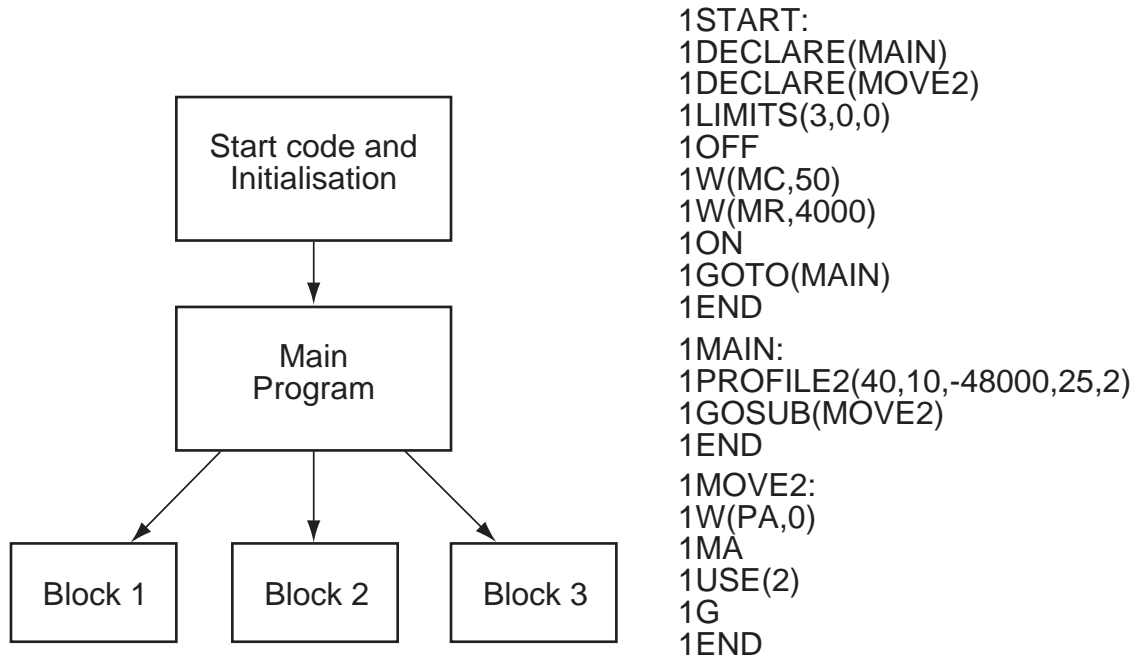


Figure 4-1. Program Structure

Example:

1START:	; start label definition
1DECLARE(MAIN)	; declare labels
1DECLARE(MOVE2)	; declare move 2
1LIMITS(3,0,0)	; configure limits.
1OFF	; remove motor power before changing motor parameters
1W(MC,50)	; set motor current to 50%.
1W(MR,4000)	; set motor resolution 4000 steps rev.
1ON	; energise motor
1GOTO(MAIN)	; goto main program
1END	

The MAIN part of the program is used to define profiles and to control the order of moves:

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

Finally individual moves are called 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 40rps² , DECELERATION 10rps² , DISTANCE 48000 steps (12 REVS MOVE), NEGATIVE DIRECTION , VELOCITY 25 rps , 2 rps START / STOP SPEED.

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, are the equivalent of sequences.

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).
1OFF                                ; remove motor power before changing motor parameters
1W(MC,50)                           ; set motor current to 50%.
1W(MR,4000)                         ; set motor resolution 4000 steps rev.
1ON                                 ; energise motor
1PROFILE1(80,20,24000,20,0)         ; 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 80rps² , DECELERATION 20rps², DISTANCE 24000 steps (6 REVS MOVE), POSITIVE DIRECTION , VELOCITY 20 rps , 0 START / STOP SPEED.

LOOP Command

The block structure of the code lends itself to performing repetitive operations, using the LOOP command. A particular labelled block of code can be called a specified number of times. Such code could be used for intermediate repetitive processes, such as marking the surface of each work piece at some point in the production cycle.

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)         ; configure limits
1OFF                   ; remove motor power before changing motor parameters
1W(MC,50)              ; set motor current to 50%.
1W(MR,4000)            ; set motor resolution 4000 steps rev.
1ON                    ;energise motor
1PROFILE3(100,50,4000,35,1) ; 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(XXX0XXXX)          ; ensure o/p 4 is off
1T1                    ; wait for 1 sec delay
1G                     ; execute move
1O(XXX1XXXX)          ; turn on o/p 4
1T1                    ; wait for 1 sec delay
1END                   ; end of label definition

```

Reserved System Labels

Certain labels have been pre-defined and 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

The pre-declared label named FAULT is used 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 event triggered code
- ARM must be set to enable a FAULT label

This means FAULT label code must be present and the **ARM11** 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 decision 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 decision 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	Main ARM bit	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	Y	N/A	Y	Y	Y	Y	N/A
Hit limit	Y	Y	Y	Y	Y	Y	Y	N/A
Drive fault	Y	Y	N/A	Y	N/A	N/A	N/A	Y
GOTO	Y	N/A	N/A	N/A	N/A	N/A	N/A	Y
GOSUB	Y	N/A	N/A	N/A	N/A	N/A	N/A	Y
LOOP	Y	N/A	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(1XXXXXXX)          ;turn on output 1 - drive OK
.
<main process commands>
.
1END

1FAULT:               ;fault routine
1O(0XXXXXXX)          ;turn off output 1 - drive fault
.
<diagnostic code - if required>*
.
1TR(IN,=,1XXXXXXX)    ;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.

Event Driven Code

Event driven code is code that is executed when a particular event occurs, such as a fault, a registration mark being detected or an input changing state. The event effectively interrupts the drive's 'normal' operation and, under certain circumstances, can immediately jump to a defined label and start executing the code following that label. Its use allows a much faster reaction time to external signals than can be obtained using conventional polling techniques.

Event driven code is enabled or disabled using the ARM command. The command allows you to control the way the drive responds to events, although certain incidents such as hitting a limit (if enabled) will always be actioned regardless of the state of the ARM command.

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 ARM1 set. If you save a program with ARM0 set, the start-up code will not run and the indexer will only respond to serial input commands.

*Unless a drive fault is pending and a fault routine is defined and armed.

Input Events

You can let user inputs trigger events by the use of labels 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(00000000)     ;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: Event triggered code execution occurs when the indexer detects an event for which a label exists.

The occurrence of an event trigger will cause the program code to immediately jump to the pre-defined input label and perform the code following that label. When the END label of this code is reached the program flow will not return to the code it was executing just before the event took place, but will continue to scan the inputs for another trigger event.

When using event triggers you must be aware that altering any basic operating parameters, such as velocity, in the event code will change the value being used in subsequent events. Consequently, you will need to fully define the move required in each event code block. This is readily accomplished by the use of USE commands to select a particular default value.

System Variables

System variables are named variables held within the drive's indexer that are used for storing a variety of system values and settings. All system variables can be read using the Report System Parameter (R command), but only certain variables can be written to using the Write (W command).

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 system variables by delaying code execution until the value of a system variable matches some stored number or string within the program.

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
BU	Buffer usage	Y	N	% of program buffer used 0-100
CQ	Command queuing	Y	Y	1= Pauses until move complete (default) 0= continuous execution
DF	Drive Fault status	Y	N	See following section
EI	Encoder Input	Y	Y	N/A
EO	Encoder signal output	Y	Y	N/A
EP	Encoder Position	Y	Y	N/A
ER	Feedback encoder resolution	Y	Y	N/A
EX	Comms. Response Style & echo control	Y	Y	0= speak when spoken to, echo off, RS485, multi-axis 1= speak whenever, echo off, RS485, single axis 2= speak when spoken to, echo on, RS232, multi-axis 3= speak whenever, echo on, RS232, single axis
IN	Inputs	N	N	Same format as IS command
IP	In Position flag	Y	N	1= In position or 0= not yet in position
IT	In Position Time	Y	Y	1-500mS, default=10mS
MC	Motor current	Y	Y	50, 60, 70, 80, 90, 100 (in percent), default 50
MR	Motor resolution	Y	Y	400, 800, 2000, 4000, default 4000
MS	Motor standby current	Y	Y	50, 70 (in percent) , default 50
MV	Moving	Y	N	Flag 1= moving or 0 = not moving
PA	Position Absolute	Y	Y	-2,147,483,648 to 2,147,483,647 steps/counts, default 0
PI	Position Incremental	Y	Y	-2,147,483,648 to 2,147,483,647 steps/counts, default 0
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
SN	Serial number	Y	N	Drive serial number
ST	Status of indexing	Y	N	See following sub-section
UF	User program Fault status	Y	N	See following sub-section

Table 4-2. List of System Variables

Position reporting

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 motion is taking place. The IP flag can only go high once movement has stopped and the IT timer value has timed-out. In Position Time (IT) allows time for mechanical oscillations to settle before indicating that the motor is in position.

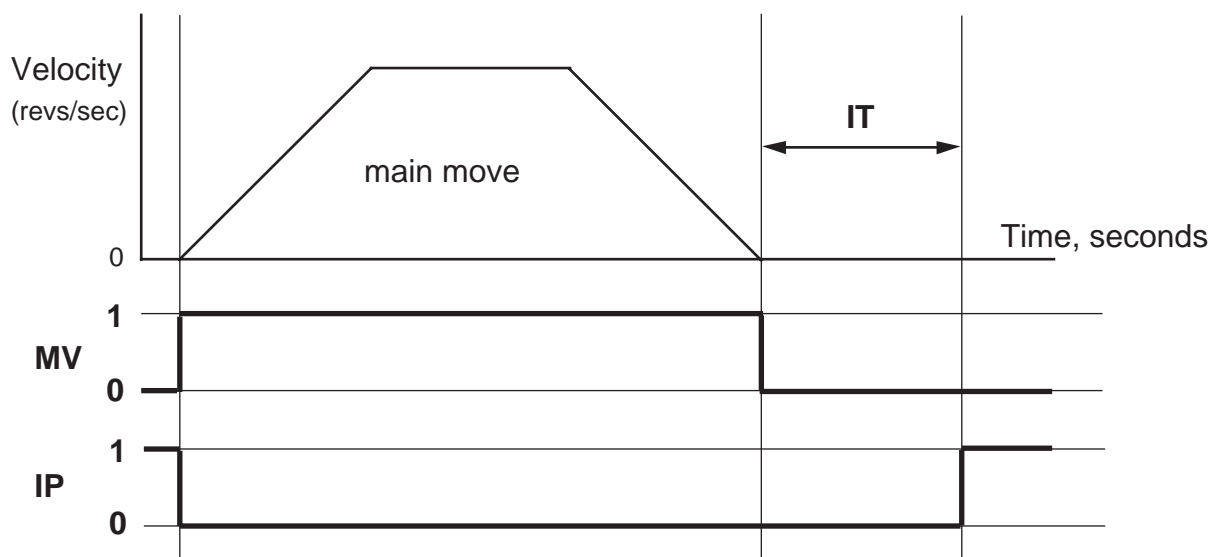


Figure 4-2. Interaction of MV, IP and IT

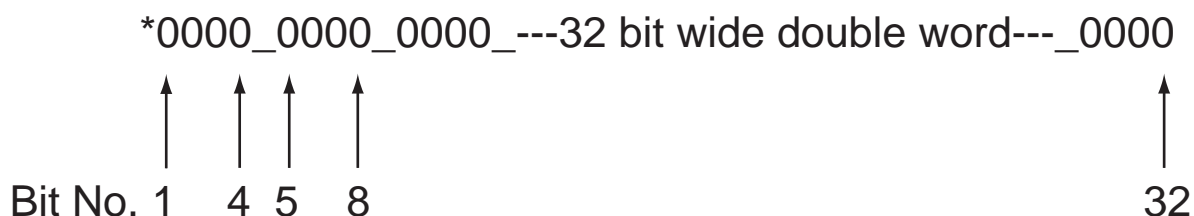
Reporting the Status of Variables

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 4-3 to determine the Status Information being reported. Use the Read command to display the ST word pattern, that is '**aR(ST)**'.

Bit Number	Status Information
1	Command processing paused
2	Looping (command executing)
3	Wait for trigger (input)
4	Running program
5	Going home
6	Waiting for delay timeout
7	Registration in progress
9	Motor energised
11	Event triggered - active until trigger inputs are reset
12	Input in LSEL not matching label
13	-ve limit seen during last move
14	+ve limit seen during last move
19	Moving (in motion)
20	Stationary (in position)
21	No registration signal seen in registration window
22	Cannot stop within the defined registration distance
23 to 32	Reserved

Table 4-3. Status Bits Description (continued)

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

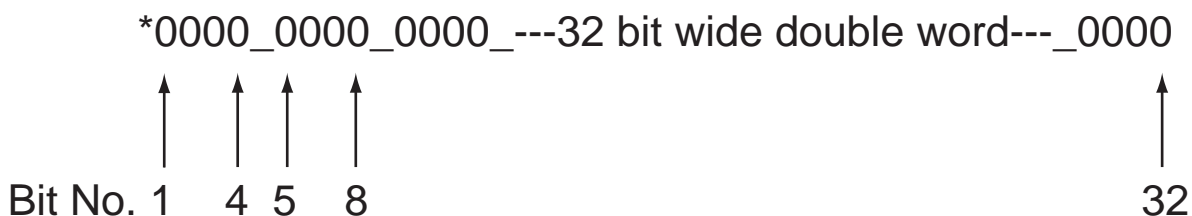
Faults are classified into two groups:

- 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 4-4 to determine the Drive Fault being reported. Use the Read command to display the DF word pattern, that is '**aR(DF)**'.

Bit Number	DF Information
1	Composite fault
2	Output stage over current
3	Supply rail failure
4	Reserved
5	Drive over temperature
6	Configuration error
7	Motor high voltage rail failure
8	Output fault
9 to 32	Reserved

Table 4-4. 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 4-5.

Bit Number	UF Information
1	Value is out of range
2	Incorrect command syntax
3	Last label already in use
4	Label of this name not defined
6	Homing failed - no signal detected
7	Home signal too narrow
8	Drive de-energised
9	Cannot relate END statement to a label
10	Program memory buffer full
11	No more motion profiles available
12	No more sequence labels available
13	End of travel limit hit
14	Still moving
15	Deceleration error
16	Transmit buffer overflow*
17	User program nesting overflow
18	Cannot use an undefined profile
19	Drive not ready
22	Save error
23	Command not supported by this product
24 to 32	Reserved

Table 4-5. 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 500rps ² 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 4-6. Where the variable can also be used with the TR command a 'Y' appears in the TR column.

Variable	Name	>	<	=	<>	TR	Format
DF	Drive fault status	N	N	Y	Y	Y	binary
IN	Inputs	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
PI	Position incremental	Y	Y	Y*	Y	Y	decimal
RM	Registration move	N	N	Y	Y	N	bit
ST	Status of indexing	N	N	Y	Y	Y	binary
UF	User program fault status	N	N	Y	Y	N	binary

* Not recommended during motion

Table 4-6. 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,=,XXXXXX1XX)                ;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 rotary 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 4-3 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

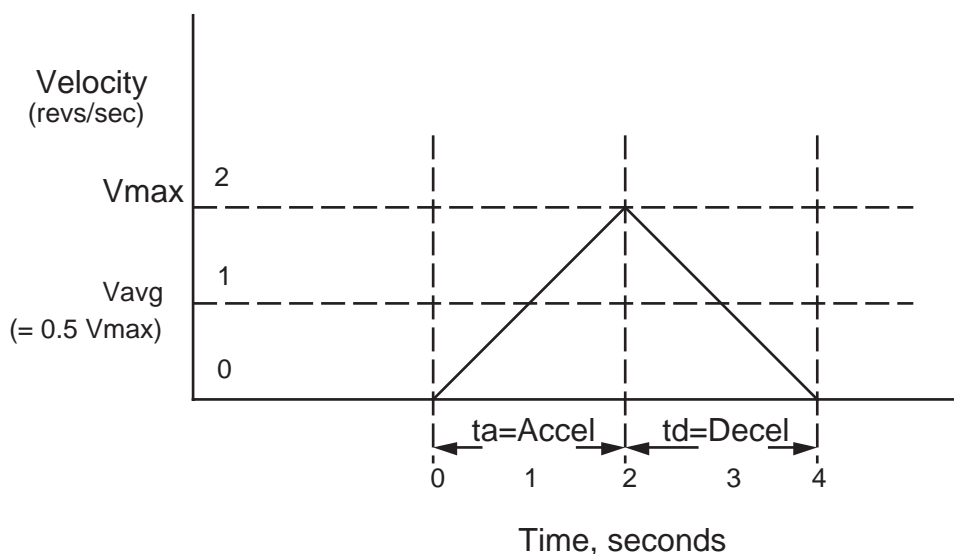


Figure 4-3. Triangular Profile

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/ sec², velocity is set to 1 rev/sec, and distance is specified as 20000 steps (5 revs), the resulting motion profile would look like this:

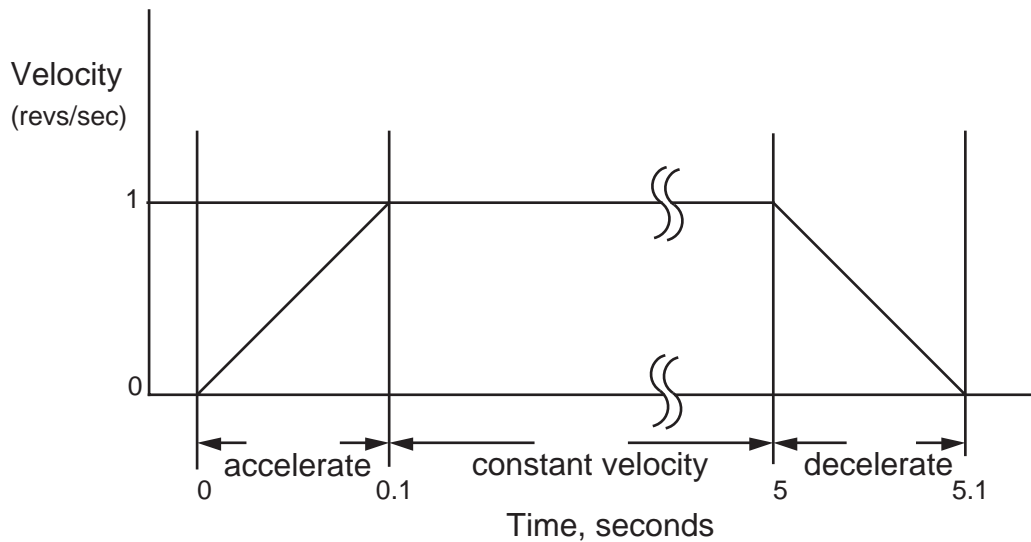


Figure 4-4. Trapezoidal Profile

Point to Point Positioning

Point to point positioning is often required in pick-and-place applications where a series of x,y co-ordinate moves needs to be performed quickly and to high accuracy. This type of application can greatly benefit from the use of an instantaneous start/stop speed. By starting a stepper motor at a pre-defined velocity rather than letting it 'ramp' up to speed can save a considerable amount of move time within repetitive point to point positioning moves. In a similar manner, time may also be saved by stopping the motor immediately rather than allowing the speed to 'ramp' down to zero.

The trapezoidal profile often used for point to point positioning can be modified to include instantaneous start/stop, resulting in the profile shown in Figure 4-5.

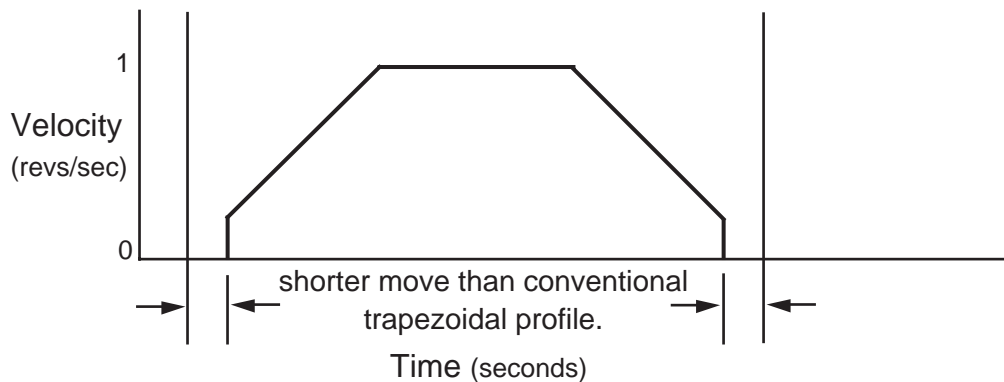


Figure 4-5. Modified Trapezoidal Profile

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**) and armed (**1ARM1**), 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 8). The default value is 0 (no output).

Care must be taken to make sure the registration move can be performed in the time available, taking into account the velocity of the initial move and the characteristics of the required registration move. If insufficient time is allowed, the registration move may appear to be performed, but the NOREG label is executed.

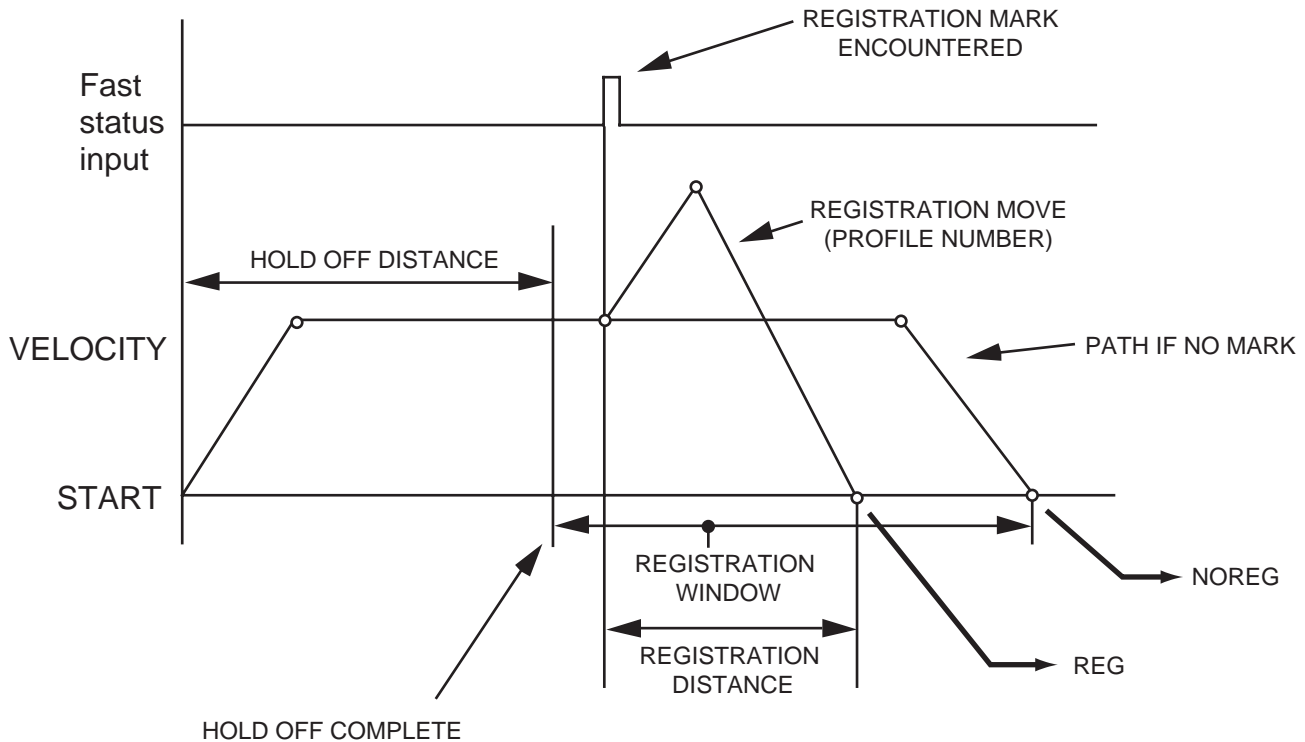


Figure 4-6. Registration Move Profile

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:

Note: you must arm a registration move.

1START:	;start label definition
1PROFILE4(10,10,40000,5,1)	;define move parameters
1PROFILE5(20,20,10000,10,5)	;define move parameters
1ARM1	;arm the registration move
1REG1(1,5,5000)	;define registration move parameters

```

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

1REG:            ;on reg mark valid turn on o/p 8 (batch counter)
1O(XXXXXXX1)
1T0.5            ;wait for 500ms delay
1O(XXXXXXX0)    ;turn off o/p 8
1END            ;end of label definition

1NOREG:         ;if reg mark not valid/seen
1O(XXXXXXX1X)   ;turn on o/p 7
1T0.25          ;wait for 250ms delay
1O(XXXXXXX0X)   ;turn off o/p 7
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.

Another feature of homing, is the ability to seek the nearest drive zero phase position, which also can improve the homing repeatability.

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

Zero phase - a defined current state in the motor which results in one of 50 repeatable shaft positions

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

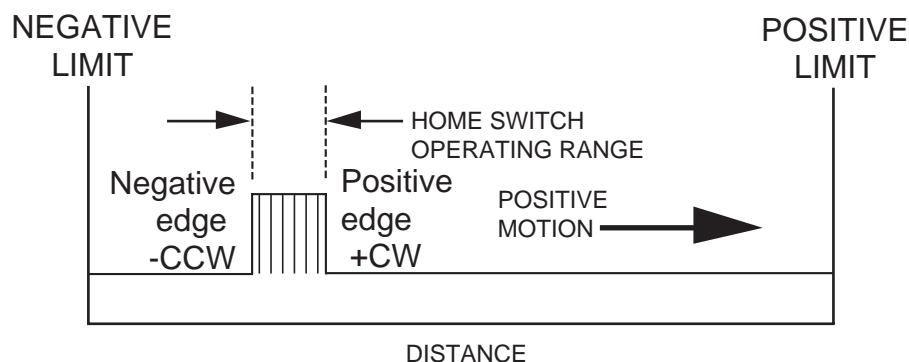


Figure 4-7. Home Switch Operation

Switch Considerations

Any type of switch will have an operating range that may correspond with a significant rotation 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. To improve repeatability still further, the edge may be detected in combination with the zero phase of the motor.

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 HOME is used to define the absolute position of the indexer, that is system variable PA is set to zero at the home position. 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 defined 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 - will behave the same as mode 1, but the indexer will also take account of the nearest drive zero phase position and will stop at that phase position

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. To reduce this switch operating point variation modes 2 and 3 can be used.

Mode 2 operation requires the home switch to operate reliably and consistently in less than the distance separating two zero phase pulses, otherwise the home position will be inconsistent. For a 200 step/rev motor zero phase will occur 50 times per rev so the gearing between the motor and load must be arranged to allow unambiguous detection of the same zero phase occurring after the required home edge is detected. To aid the setup of an application using zero phase home detection, the HV LED will illuminate with an amber coloured light at every zero phase position of the motor.

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 of 0.1 rps (start/stop range of motor) 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, in which case a warning 'home switch too narrow' will be reported, but homing will continue from the other side of the home switch operating range.

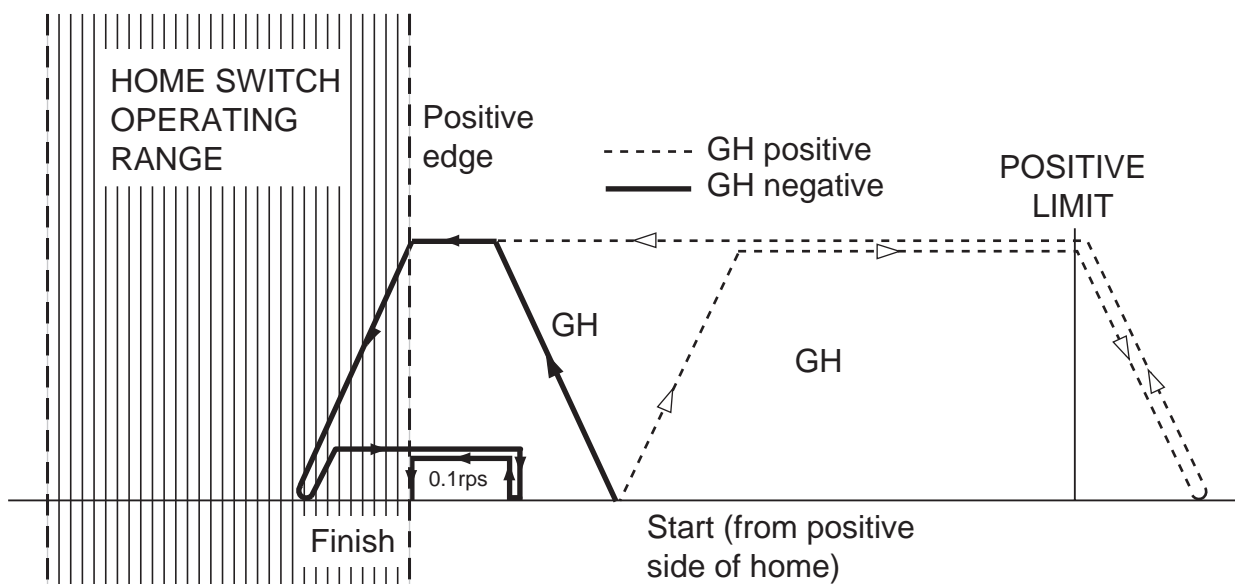


Figure 4-8. Go Home to Positive Edge

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.

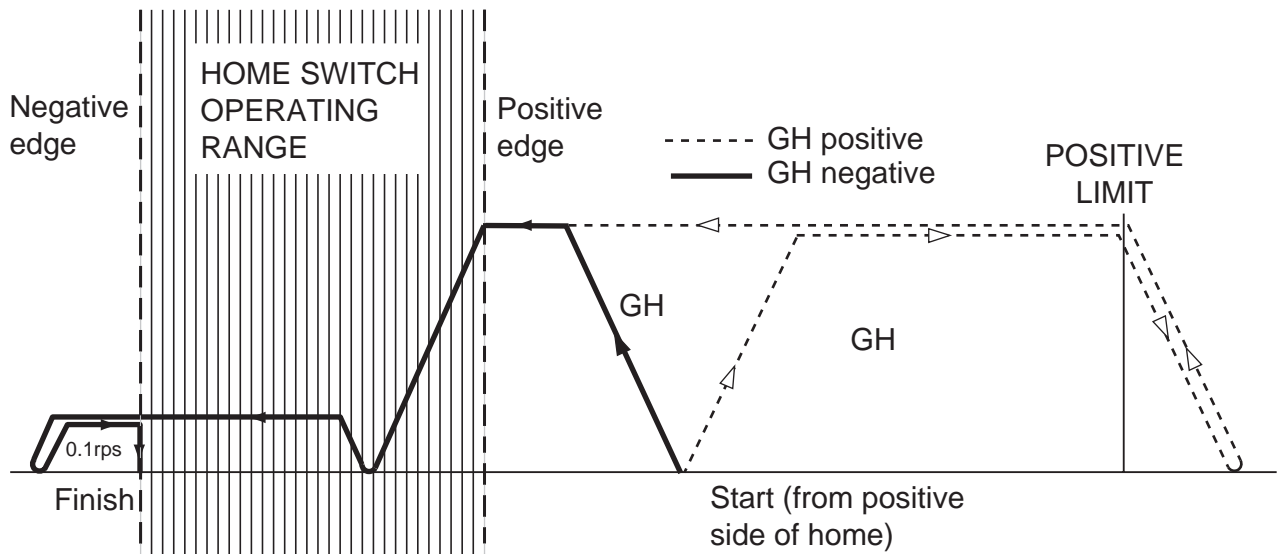


Figure 4-9. Go Home to Negative Edge

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.

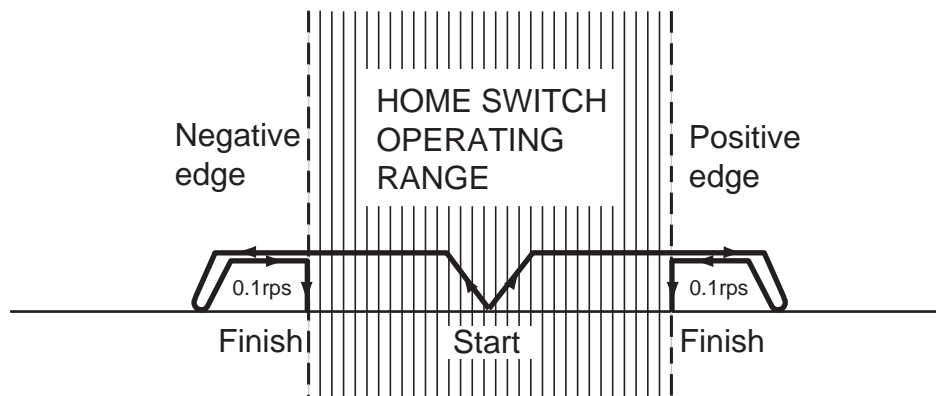


Figure 4-10. Go Home Starting from Home

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.

Note: you must arm a home move.

Example of Homing (Datum) Routine

```
1START:                ; start label definition
1ARM1X                 ; arm the home command
1DECLARE(MOVE3)        ; declare label
1LIMITS(0,1,0)        ; configure limits (enabled, normally closed, stop when hit).
1OFF                   ; remove motor power before changing motor parameters
1W(MC,50)              ; set motor current to 50%.
1W(MR,4000)            ; set motor resolution 4000 steps rev.
1ON
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 5 & 6) 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 (**ARM11**)
- 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.

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.

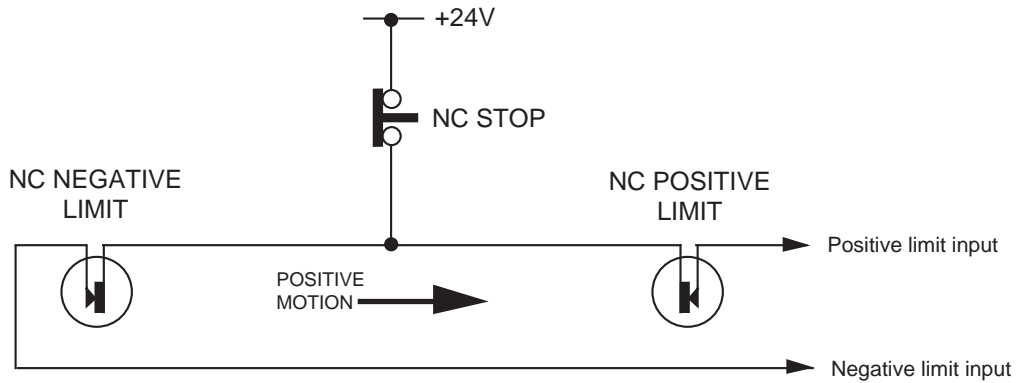


Figure 4-11. Limit and Stop Switch Configuration

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.

Stop Input

A Stop function can be implemented based upon the scheme shown in Figure 4-11. Whilst motion is in progress, momentarily pressing the STOP button will take both limit inputs 'low' which will stop the motor at the limit deceleration rate. To use this form of hardware stop, both limits will need to be enabled in the LIMITS command with the type field set to '1' (default).

Pressing the STOP button will activate both limits at once, this will cause motion to halt and no more movement will take place until both limits have been cleared. The status will report the first limit seen, but if both limit switches operated within the same millisecond period the positive limit will be reported as being active.

Following a hardware stop being actioned, you should check to make sure it is safe to continue motion and to resume sending commands to the drive.

Note: As this stop function relies upon correct software operation, it is not appropriate to rely on the stop input as providing a safety function.

5. EASI-Tools Software

Computer Requirements

To be able to run EASI-TOOLS software, necessary for the control and programming of the XL25i/XL50i/L80i, you will require an IBM™ compatible PC running Windows 95/98™ or NT4. A Windows 3.1/3.11™ version is available upon request.

The PC needs to be specified to run Windows 95™ with at least 16MB of RAM, a VGA monitor and a Windows™ compatible mouse. The installed program size is approximately 800K.

Serial Link Lead

If the computer being used is fitted with 9-way D-type COM connectors, you will need a **2-wire plus ground** 'RS232 style' lead which has the **Rx and Tx wires crossed over**. Wiring details are given in the ***Electrical Installation*** section. If a 25-way communications port is being used refer to the Installation section for details of plug wiring.

Stepper Motor

Both drives are designed to be used with **general purpose stepper motors up to a frame size of 34**. Provided the motor's minimum inductance is within the range quoted in the Specification and it will not be damaged by the current supplied by the drive, any series or parallel connected motor may be used.

Note: The information contained within this section applies to EASI-TOOLS software version 1.5 or greater and XLi drives with firmware 2.18 or greater. If you have an earlier version of EASI-TOOLS 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. Installed space is less than 1MB.

Once software installation is complete, commands can be downloaded from the PC to the drive to confirm its operation. If EASI-TOOLS has already been loaded you may skip the following Installation and Operation sections up to ***Confirming Drive Operation***.

Software Installation

Before attempting to install the EASI-TOOLS software supplied with your drive check that your PC meets the requirements previously defined under ***Computer Requirements***. EASI-TOOLS software is supplied on one 3.5 inch diskette and installs in the usual manner common to Window™ applications.

Installation Procedure

This procedure takes you quickly through the steps necessary to install EASI-TOOLS 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-TOOLS software follows:

1. Place the EASI-TOOLS installation disk in drive A.
2. Click on the Windows™ START BUTTON and select menu option RUN. In the command line type: **a:\setup** and press return.

Note: If you do not have the menu option RUN, navigate to drive A: and double click the **setup.exe** file.

3. The screen will display the EASI-TOOLS program banner and will prepare an installation setup program.
4. The banner screen is automatically replaced by a Welcome dialog 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 dialog box giving you the options of Exit Setup, which returns you to Windows™ or Resume which takes you back to the Welcome box.
5. Selecting NEXT> displays a 'Choose Destination Location' dialog/selection box which provides the option of installing EASI-TOOLS in the directory of your choice. The default directory is **c:\Parker\EASITool**, other buttons are described within the dialog box, see Figure 5-1.

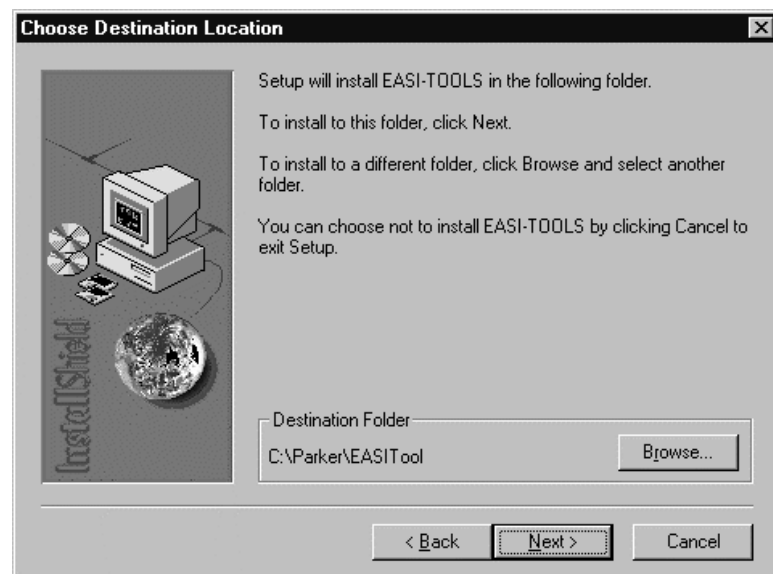


Figure 5-1. Choosing Where to Install EASI-TOOLS

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

7. Once EASI-TOOLS has been loaded, the screen will display a message dialog 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-TOOLS may also be run from the Start menu.



Figure 5-2. EASI-TOOLS Application Window

Uninstalling EASI-TOOLS

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

Software Operation

Once installed, EASI-TOOLS can be started from the start menu or by double clicking its application icon. At startup EASI-TOOLS shows the version number being used.

The main application window is displayed, entitled 'Parker Hannifin EMD - EASI TOOLS', 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

<u>N</u> ew	Ctrl+N
<u>O</u> pen...	Ctrl+O
<u>S</u> ave	Ctrl+S
Save <u>A</u> s...	
<u>P</u> rint...	Ctrl+P
<u>C</u> lose Window	
<u>E</u> xit	Alt+F4

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 specifying the file name

Print the editor file or contents of terminal buffer

Close current active window

Exit EASI-TOOLS

Edit

<u>U</u> ndo	Ctrl+Z
<u>C</u> ut	Ctrl+X
<u>C</u> opy	Ctrl+C
<u>P</u> aste	Ctrl+V
<u>C</u> lear	Del
<u>S</u> elect All	
<u>G</u> o to line	Ctrl+G

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

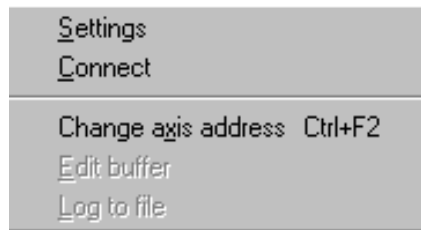
<u>F</u> ind...	
Find <u>N</u> ext	F3
<u>R</u> eplace...	

Editor search & replace operations

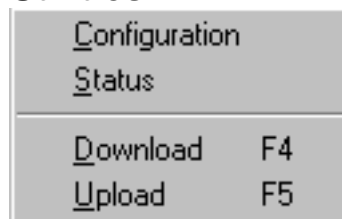
Find specified text (top down from cursor position)

Repeat search again

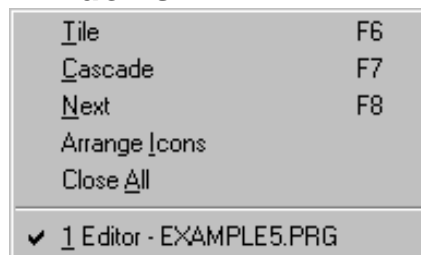
Find and replace text (top down from cursor position)

TerminalTerminal on-line operations

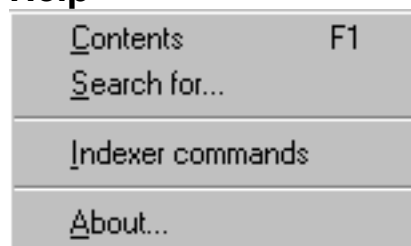
Configure the serial communications
 Open/close the terminal (after testing the connection)
 Change address on multi axis systems
 Create an editor file window or terminal buffer
 Open/close file logging terminal buffer actions

UtilitiesSpecific tools

Setup drive for operation
 Report of user drive faults and status
 Transmit current active editor contents to drive
 Upload drive program(s) to new editor file window

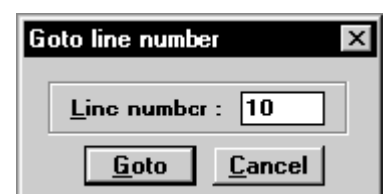
WindowsWindow controls

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

HelpProgram help facilities

Open help file at the main contents (start)
 Prompt for topic string and search help file
 Open help for individual EASI commands
 EASI-TOOLS version number and copyright

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



Communicating with a Drive

This sub-section assumes a correctly wired communications link has been established between the drive and PC and the required address settings have been made.

WARNING

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

Configure the serial communications

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

The default settings used are:

Port	COM2
BAUD rate	9600
Options	Auto wrap

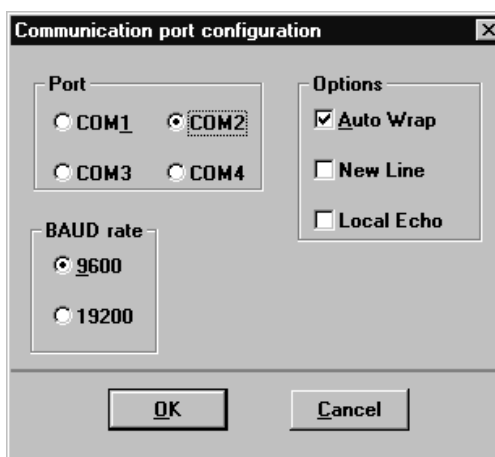


Figure 5-3. Comms Port Configuration

Once the required configuration has been selected, 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 indexer' 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-TOOLS installation package to give you a start with drive programming.

Note: If you are using a PC with a serial mouse avoid IRQ conflicts between serial ports by selecting a serial port with a different IRQ to that used by your mouse. Comm ports 1 and 3 share the same IRQ as do ports 2 and 4. Usually, if your mouse is connected to COM1, connect the drive to COM2.

Change Axis Address

The Change axis address menu option allows you to change axes in a multi axis system. Selecting the menu option displays the dialog box shown in Figure 5-4.



Figure 5-4. Change Axis Address Tool

Drive Configuration

Assuming successful communications have been made, you can now begin to configure your drive using the Configuration dialog box selected from the Configuration option of the Utilities menu. This box is arranged as a series of tabs giving you access to six configuration options, as shown in Figure 5-5. A pull-down numbered list, allows address selection. If a mix of drive types is being used, EASI-TOOLS will automatically identify the drives present and will grey-out options not available for a particular drive type. This means the encoder option cannot be selected for an XLI drive.

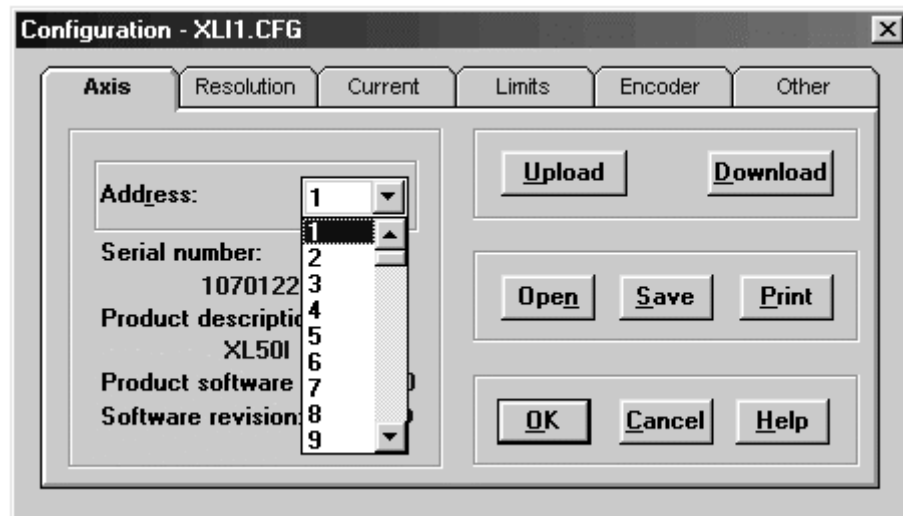


Figure 5-5. Drive Configuration Settings

The right hand side of this box is common to all tab options and will only be described once under Axis Configuration.

Axis Configuration

This window allows you to address a particular drive having the axis address (set within the drive) displayed in the pull-down numbered list address box. When different axes are addressed, the software revision and serial number of the drive being addressed will be displayed.

Download Button

When all of the configuration information has been set for every tab position the Download button is used to transfer configuration information to the drive on the presently addressed axis.

Upload Button

Activating this button will upload configuration information from the currently addressed drive to the EASI-TOOL screens, allowing a particular axis configuration to be examined.

Configuration Files

The three buttons Open, Save and Print are all associated with drive configuration files, that allow the configuration of a particular drive to be saved to disk as a .cfg file.

Open

Allows you to search for a particular configuration file that may be opened, edited, downloaded or saved.

Save

Allows the current configuration information to be saved to file in a directory of your choice.

Print

Allows the current configuration file to be printed. The printout lists axis details, motor setup, end of travel limits and other parameters.

OK

Is selected when you have finished making changes to the configuration.

Cancel

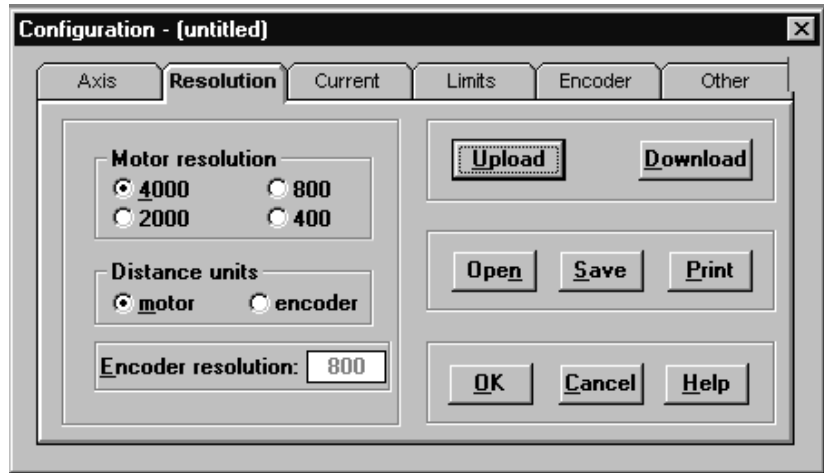
Is used to de-select the configuration dialog box.

Note: when you select either OK or Cancel and changes have been made to the configuration, you are prompted with a message asking if you wish to save the changes to disk.

Other Configuration Options

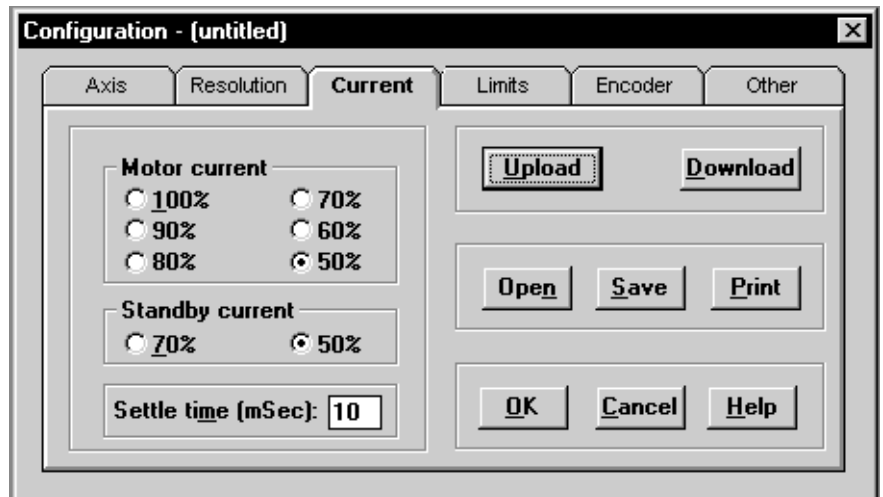
The following configuration screens illustrate the range of configuration options possible within EASI-TOOLS.

Resolution Configuration



The 'Resolution Configuration' dialog box is titled 'Configuration - (untitled)'. It features a tabbed interface with 'Resolution' selected. The 'Motor resolution' section has four radio buttons: 4000 (selected), 2000, 800, and 400. The 'Distance units' section has two radio buttons: motor (selected) and encoder. The 'Encoder resolution' is set to 800 in a text field. On the right, there are buttons for 'Upload', 'Download', 'Open', 'Save', 'Print', 'OK', 'Cancel', and 'Help'.

Current Configuration



The 'Current Configuration' dialog box is titled 'Configuration - (untitled)'. It features a tabbed interface with 'Current' selected. The 'Motor current' section has six radio buttons: 100%, 90%, 80%, 70%, 60%, and 50% (selected). The 'Standby current' section has two radio buttons: 70% and 50% (selected). The 'Settle time (mSec)' is set to 10 in a text field. On the right, there are buttons for 'Upload', 'Download', 'Open', 'Save', 'Print', 'OK', 'Cancel', and 'Help'.

Limits Configuration

The screenshot shows the 'Limits' tab of the 'Configuration - (untitled)' window. The 'Mask' section has four radio buttons: 'Enable both' (selected), 'Disable limit+', 'Disable limit-', and 'Disable both'. The 'Switch normally' section has two radio buttons: 'Closed' (selected) and 'Open'. The 'Limit mode' checkbox is unchecked. The 'Deceleration (rps/s)' text box contains the value '200.0'. On the right, there are buttons for 'Upload', 'Download', 'Open', 'Save', 'Print', 'OK', 'Cancel', and 'Help'.

Encoder Configuration

NOT AVAILABLE FOR XLI

The screenshot shows the 'Encoder' tab of the 'Configuration - (untitled)' window. The 'Encoder inputs style' section has two radio buttons: 'Step/Dirn' (selected) and 'A B Z'. The 'Encoder outputs style' section also has two radio buttons: 'Step/Dirn' (selected) and 'A B Z'. On the right, there are buttons for 'Upload', 'Download', 'Open', 'Save', 'Print', 'OK', 'Cancel', and 'Help'.

Other Configuration

The screenshot shows the 'Other' tab of the 'Configuration - (untitled)' window. The 'Programming' section has 'Motor energized' (unchecked) and 'Command queuing' (checked). The 'ARMing' section has 'Events enabled' (unchecked) and 'FAULT label' (checked). The 'Communications style' section has 'Speak whenever' (checked) and 'Echo-back ON' (checked). On the right, there are buttons for 'Upload', 'Download', 'Open', 'Save', 'Print', 'OK', 'Cancel', and 'Help'.

Status

The Utilities menu option Status provides a convenient method of examining the double word

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. The usual communication checks will be made and if all is OK the program will be downloaded to the drive's internal memory. A shortcut download is provided by function key F4.

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 dialog 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.



Confirming Drive Operation

With the drive and motor correctly wired and the serial connection made to a PC running EASI-TOOLS 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.

```
1START:           ; define start label code
1ON               ; enable the drive
1LIMITS(3,0,0)    ; disable limits
1D4000            ; set distance to 4000 steps (1 rev)
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 1 revolution 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 been given a simple 1 to 7 character name, usually an abbreviation of its full descriptive title. Commands are listed in alphabetic order with any non-alphabetic symbols appearing last.

Each individual description will include a one line header giving the abbreviated name

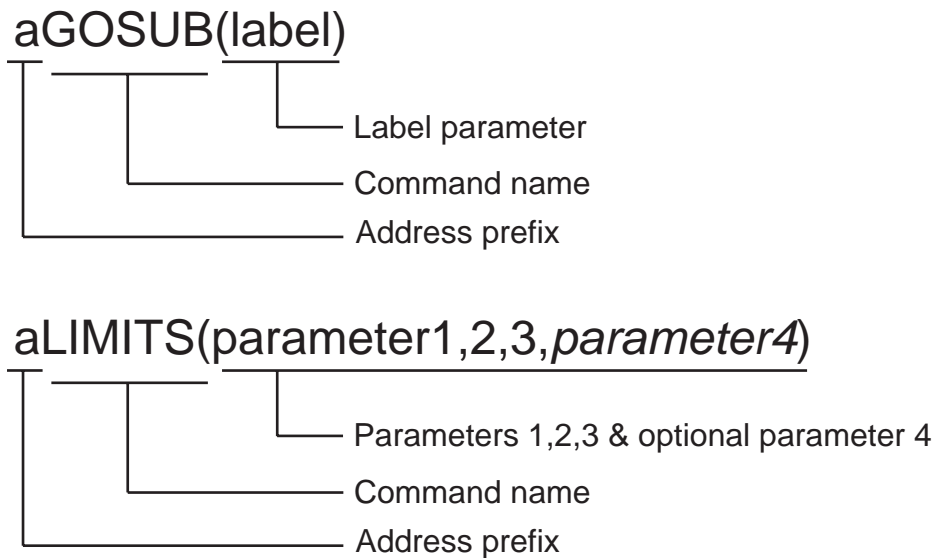


Figure 6-2. Parameter Commands

Attention

- [1] Commands can only be terminated 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-TOOLS, that is they contain comments and line feeds etc.

Command Properties

Each command has a particular set of properties which 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)

These commands are acted upon by the indexer 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, therefore such commands as 1IS, 1R(ST), 1R(UF), 1A etc. are correctly reported. 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

1TR(IN,=,X0X0111111)	;input trigger false
1R(MC)	;read motor current
*50	
1IS	;read input status
*0111100010	;virtual inputs 9 & 10 reflect the state of 5 & 6
1A50	;send buffered command
1R(MR)	;report commands are now delayed until the trigger
1IS	;command is complete
.	
.	
(Trigger becomes true, 1A50 actioned)	
*4000	;report commands completed, MR=4000
*0000111111	;new input status reported

Can be used in labelled block

Means the command can be saved within a labelled program block. In this case the command will only be executed when that labelled block has been requested. Commands used within the definition of a labelled block are only saved when SV is sent.

Can't be used in labelled block

Means the command cannot be saved 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 no data is changed by a command, 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 on entry for syntax and parameter limits. 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 message will be reported. 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 aAn	Units Revs/sec ²	Range of 'n' 0.1 to 1023.9	Default 10	See also AA AD
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, if previously set, will be over written.			
Properties	Immediate or buffered, can be used in labelled block, saved by SV			
Example	<p>To set the acceleration and deceleration rates of axis 1 to 120 rps², type 1A120</p> <p>To determine the acceleration of axis 1, type 1A</p> <p>The response is *120.0 120.0</p> <p>Overrange value 1A1050</p> <p>Will be reported as..... *E (meaning error)</p>			
Note	In a positioning move the deceleration time is limited to 65 seconds. If a very low value for deceleration is used together with a high peak velocity such that the deceleration time is greater than 65 seconds, then an error will be reported when G is issued.			

AA Acceleration

Syntax aAAAn	Units Revs/sec ²	Range of 'n' 0.1 to 1023.9	Default 10	See also A AD
Description	The AA command will set or report the programmed linear acceleration rate of the motor. The acceleration value assigned to the AA command will be over written by any following A command.			
Properties	Immediate or buffered, can be used in labelled block, saved by SV			
Example	<p>To set the acceleration and deceleration rates of axis 1 to 120 rps², type 1AA120</p> <p>To determine the acceleration of axis 1, type 1AA</p> <p>The response is *120.0</p> <p>Overrange value 1A1024</p> <p>Will be reported as..... *E (meaning error)</p>			

AD Deceleration

Syntax aADn	Units Revs/sec ²	Range of 'n' 0.1 to 1023.9	Default 10	See also A AA
Description	The AD command will set or report the programmed linear deceleration rate of the motor. The deceleration value assigned to the AD command will be over written by any following A command.			
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..... 1A1040 Will be reported as *E (meaning error)			
Note	In a positioning move the deceleration time is limited to 65 seconds. If a very low value for deceleration is used together with a high peak velocity such that the deceleration time is greater than 65 seconds, then an error will be reported when G is issued.			

ARM Enable event triggered code

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

The **ARM** command allows you to enable (arm) or disable (disarm) the following commands: **HOME**, **LSEL** & **REG**. It also enables/disables the fault label.

n=1 : arm
n=0 : disarm (default condition)

The second parameter 'm' is a fault switch that enables/disables the fault label from being run.

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

All individually armed commands can be globally disarmed using only one command and can then be re-armed where required.

At power on, when saved and armed, the indexer 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
1ON
1LIMITS(3,0,0)                        ;disable limits
1PROFILE1(100,100,4000,25,0)          ;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 indexer with **ARM0**, then the start-up sequence will fail to run, and the indexer will wait for serial commands.

Requesting an ARM status will report the state of all the commands that may be armed or disarmed, for example:

```
aARM
*ARM 1
*HOME 1
*LSEL 1
*REG 0
*FAULT 0
```

C Continue

Syntax aC	Units -	Range of 'n' -	Default -	See also PS
Description	The C (continue) command causes a user commands 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 100rps ² 1V20 ;velocity 20rps 1G ;go 1"*TEST" ;add TEST comment 1C ;continue *TEST ;message TEST is displayed			

CLEAR Clear user code

Syntax aCLEAR(label)	Units -	Range of 'n' -	Default -	See also -
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	Once a label has been declared, the declaration cannot be cleared individually. You can only clear declarations by using CLEAR(ALL).			

Declare Declare

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

Description All labels, apart from START, REG, NOREG & FAULT need to be declared at the beginning of the program using a **DECLARE** command. You can declare up to 10 labels as well as the four system labels.

Labels can only be declared 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 still 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%

D Distance

Syntax aDn	Units Steps	Range of 'n' -2,147,483,648 to 2,147,483,647	Default 4000	See also M
Description	The D command will set or report the programmed move distance. The value programmed is only used for preset moves and is ignored in MC (Move Continuous).			
Properties	Immediate or buffered, can be used in labelled block, saved by SV			
Example	<p>To set the move distance of axis 2 to 15000 steps type.. 2D15000</p> <p>To report the current programmed move distance of axis 2, type 2D</p> <p>The controller responds with..... *15000</p> <p>If a value entered is out of range *E will be reported and the current value will not be altered.</p> <p>Distance reports the current direction as influenced by the H command in MI (Mode Incremental) only. For example:</p> <pre> MI ;mode incremental 1D4000 ;set distance to 4000 steps 1D ;report distance *4000 ;value reported 1H- ;change direction 1D ;report distance *-4000 ;value reported </pre>			

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	<p>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.</p> <p>To enable axis 6 to accept commands over the RS232 serial link, type 6E1</p> <p>To disables communications using the RS232 serial link, type 6E0</p>			

EXIT

Exit from loop

Syntax aEXIT	Units -	Range of 'n' -	Default -	See also Loop
Description	The EXIT command is used to terminate a loop function instantly.			
Properties	Immediate or buffered, can be used in labelled block, not saved by SV			
Example	<pre> 4START: ; signifies this is the power on sequence 4DECLARE(GRIP) 4PROFILE2(100,100,4000,25,0) ; define profile 2 4LOOP(GRIP,0) ; repeat the grip/ungrip code forever 4O(XX1XXXXX) ; set output 3 4T5 ; If we exit the loop then we pulse output 4O(XX0XXXXX) ; 3 to indicate gripper tension out of tolerance 4END ; 4GRIP: 4USE(2) ; Use motion parameters from profile 2 4G ; do the move 4O(XXX1XXXX) ; trigger the measurement 4H ; change to ungrip/grip 4T1 ; delay for 1 sec 4IF(IN,=,X0XXXXXX) ; gripper tension out of tolerance 4EXIT ; abort loop 4IF(IN,=,X1XXXXXX) ; else continue the loop 4G 4END </pre>			
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.			

G Go

Syntax aG	Units -	Range of 'n' -	Default -	See also PS S K VS
Description	Issuing a G command starts motion using the parameters specified by the V, AA, AD, D and VS 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(150,200,1500,25,1) ;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 aGH	Units -	Range of 'n' -	Default -	See also HOME S K ARM
Description	The go home command instructs the indexer to search for the home position as defined by the home input switch. For this command to function correctly, the home function must have been used to define the homing parameters. For more information see the sub-section on homing in Control of Xli Stepper Drives .			
Properties	Immediate or buffered, can be used in labelled block, not saved by SV			
Example	1START: ;START label definition 1ARM1 ;arm the command 1HOME1(+,1,-15,100,2) ;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.

Attempting to Go Home in the direction of a currently active limit will cause an error and the fault label will run, if armed.

GOSUB GO to SUBroutine

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

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 10 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(360,360,400000,20,0.5)      ; define some move profiles
1PROFILE2(360,360,400000,45,0.5)
1GOSUB(MOVE1)                          ; go do move 1 and come back
1GOSUB(MOVE2)                          ; go do move 2 and come back
1O(XXXXXX11)                          ; set outputs 7 and 8
1END

1MOVE1:
1USE(1)                                ; use the move profile 1
1G
1O(XXX1XXXX)                           ; turn output 4 on
1T0.1                                  ; wait for 100mS
1O(XXX0XXXX)                           ; turn output 4 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	<pre> 2DECLARE(MOVE1) ; declare move 1 2DECLARE(MOVE2) ; declare move 2 2START: ; code run after power on 2PROFILE1(360,360,400000,20,0.5) ; define some move profiles 2PROFILE2(360,360,400000,45,0.5) 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 </pre>			
Note	<p>If a GOTO command is used in conjunction with a GOSUB or LOOP command the number of nesting levels is set to zero.</p> <p>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.</p>			

H Change direction

Syntax	Units	Range of 'n'	Default	See also
aHn	-	+ - or blank	+	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 and is ignored.

In Mode Continuous, only the H command can set the direction of motion as the D command is ignored for continuous moves.

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

```
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
```

Refer to
LOOP

```
3START:
3PROFILE1(360,360,40000,20,0.5) ; 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 called, apart from when in mode continuous*. 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.

*In mode continuous, moves are always made with the direction being decided by the H command.

HOME

Configure Homing

Syntax	aHOMEon/off(reference_edge,home_type,direction_&_velocity, acceleration/deceleration,mode)
Description	<p>The Configure Homing command is used to setup homing prior to the use of the GH (Go Home) command. Before homing can be used the home input must be turned ON (ON:1, OFF:0).</p> <p>The reference edge parameter is used to select the required edge of the home switch regarded as home (+ for positive edge or - for negative edge), see Homing section.</p> <p>The home_type parameter is used to select the type of switch to be used for homing, the choice is:</p> <div><div>Home switch normally closed</div><div>1</div><div>Home switch normally open</div><div>0 (default)</div></div> <p>Velocity and direction determines the velocity at which homing is performed and the direction in which home is initially searched for. Whilst acceleration/deceleration sets the acceleration and deceleration rates used.</p> <p>The mode parameter determines what happens when the specified edge of the home switch is encountered:</p> <ul style="list-style-type: none">•mode 0 The indexer 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 The indexer is set to seek the nearest drive zero phase position, to improve homing repeatability. <p>When the home position is reached, the absolute position of the indexer is set to 0. (system variable PA). The incremental position reports the distance moved to reach the home position. (system variable PI).</p> <p>Typing aHOME on its own will return the current parameter values for the nominated axis. For example:</p> <p>The command 3HOME may return:</p> <div>* AR1 E- TP1 V+10.0 A10.0 M0</div> <p>Meaning the command is armed, reference edge is negative, home type is 1, velocity is 10 rps positive, acceleration is 10 rev/s² and mode is 0.</p>

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

aHOME1

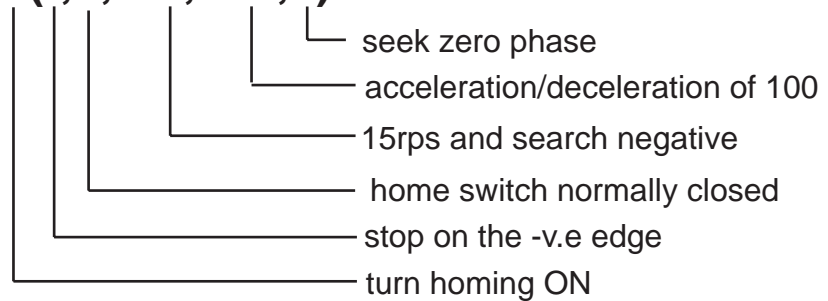
Or turned OFF using:

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 rps². The motor is to stop on the negative edge of the home switch and then seek zero phase.

3HOME1(-,1,-15,100,2)



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**alF(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
Note: input can be accepted from the keyboard/terminal.

Example

2IF(PA,=,450)	; if absolute indexer position = 450 steps on axis 2
2O(11000000)	; set outputs 1 and 2, otherwise
2IF(PA,>,500)	; if absolute indexer position > 500 steps
2O(00000001)	; set output 8

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 aIS	Units -	Range of 'n' -	Default -	See also SN
---------------	------------	-------------------	--------------	----------------

Description The **IS** command reports the status of the configurable user inputs. The response is ten (0 or 1) digits the first eight of which correspond to bits 1 to 8 in the table below (real inputs). Reported (virtual) bits 9 and 10 always follow the state of real bits 5 and 6.

X3 Pin Number	Bit	Function
2	1	User input 1
3	2	User input 2
11	3	User input 3
4	4	User input 4
12	5	User input 5/LIM-
5	6	User input 6/LIM+
13	7	Registration/ input 7
6	8	Home/ input 8
N/A	9	LIM-
N/A	10	LIM+

0 represents a low condition on the input

1 represents a high condition on the 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 *0100000100

User inputs 2 and home input 8 are high. All other inputs are low.

Note The status of inputs 5 and 6 will be mirrored on virtual inputs 9 and 10, permitting code to read limits on bits 9, 10 for compatibility with DIN Rail Li code.

K Kill

Syntax aK	Units -	Range of 'n' -	Default -	See also S, T
Description	<p>The KILL command is used to immediately stop motion. Once issued, it will cut the pulse train to the motor.</p> <p>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. This command is used as an immediate stop and does not attempt to control deceleration.</p> <p>For a controlled stop use the S (stop) command.</p> <p>The KILL command cannot be used in a label, its use is primarily for emergency situations.</p> <p>K will also terminate any program execution and disarm the event triggered code. To re-arm the event triggered code after a kill command is issued, use the ARM command.</p>			
Properties	Immediate only, can't be used in labelled block, not saved by SV			
Example	<pre>1G ;set drive in motion 1K ;stop everything</pre>			
Note	<p>The K command does require a device address or 0K to kill all axis. It will stop a time delay (T command) and will abort a program.</p> <p>This command does not replace the requirement for an additional hardware device to cut power to the motor in the event of an emergency situation arising.</p>			

LIMITS

Syntax	aLIMITS(mask,type,mode,LD)																		
Description	<p>The LIMITS command allows the user to define whether the LIM+/- inputs are being used as limit inputs or user inputs via the LIM_MASK.</p> <p>The mask field takes the following values:</p> <table> <tr><td>0</td><td>Enable limits (default setting)</td></tr> <tr><td>1</td><td>Disable limit +</td></tr> <tr><td>2</td><td>Disable limit -</td></tr> <tr><td>3</td><td>Disable limit + & -</td></tr> </table> <p>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.</p> <p>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.</p> <p>type field takes the following values:</p> <table> <tr><td>1</td><td>Limits normally closed (default setting)</td></tr> <tr><td>0</td><td>Limits normally open</td></tr> </table> <p>mode field takes the following values:</p> <table> <tr><td>0</td><td>Stop motion when a limit is hit and abort the program, then go to a predefined fault sequence, otherwise stop (default setting).</td></tr> <tr><td>1</td><td>Stop motion when a limit is hit but continue the program. In certain applications this allows the limit switch to define a home position.</td></tr> </table> <p>The optional LD parameter sets the required deceleration rate after hitting a limit, the default deceleration is 200 rps²: See deceleration command AD for range settings.</p>			0	Enable limits (default setting)	1	Disable limit +	2	Disable limit -	3	Disable limit + & -	1	Limits normally closed (default setting)	0	Limits normally open	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.
0	Enable limits (default setting)																		
1	Disable limit +																		
2	Disable limit -																		
3	Disable limit + & -																		
1	Limits normally closed (default setting)																		
0	Limits normally open																		
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.																		
Properties	Immediate or buffered, can be used in labelled block, saved by SV																		
Example	<pre> 3LIMITS(0,1,0,100) ; +ve limit enabled, normally closed switch ; -ve limit enabled, normally closed switch ; user inputs 5 and 6 unavailable for use : deceleration 100rps² 3LIMITS(1,1,0,100) ; +ve limit disabled ; user input 6 available for use ; -ve limit enabled, normally closed switch ; user input 5 unavailable for use : deceleration 100rps² </pre>																		

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 SN

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

LIST

List user program

Syntax	aLIST(<i>label</i>)
Description	The LIST command is used to view a user subroutine program in memory specified by the label parameter. This command has no effect when the source/destination is fieldbus.
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(360,360,400000,20,0.5)
1PROFILE2(360,360,400000,45,0.5)
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
```


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 allow for a sensor to measure deflection.

```
2START:                ; signifies this is the power on sequence
2DECLARE(GRIP)
2PROFILE2(150,200,4800,45,1)
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(XXX1XXXX)         ; signal grip cycle
2H                    ; change to ungrip/grip
2G                    ; do the move again
2H                    ; change direction
2T1                   ; delay for one second
2O(XXX0XXXX)         ; 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)

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.

For **LSEL** to work, the indexer must be globally armed (ARM1).

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

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

Inputs					Code type/Execution label	
1	2	3	4	5	BCD code	Binary code
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 thumbwheel.

```

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. Inputs must be present for a time exceeding the scan time, see **SN**.

See also **ARM** and **SN**

M Mode

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

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

 The values of n are:

 A - indexed move with absolute positioning

 C - continuous move

 I - indexed move with incremental positioning

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 to zero
3MA           ;mode absolute
3D1000        ;set distance
3G            ;move to absolute position 1000
3D100         ;set distance
3G            ;move to absolute position 100
3R(PA)        ;report absolute position
*100
```

To check the current mode, type..... **3M**
The response will be *MA

Note In **MC** the command **D** is ignored
 In **MA** the command **H** is ignored

O Output

Syntax aO(pattern)	Units -	pattern see below	Default 00000000	See also 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 8 bits in length in the order of outputs 1 to 8 Trailing X characters are not required.			
Properties	Immediate or buffered, can be used in labelled block, not saved by SV			
Example	2O(11001100) ; sets outputs 1,2,5,6 ON all others OFF 2O(X0XXXXX1) ; leaves outputs 1,3,4,5,6,7 as they were and ; turns output 2 OFF, output 8 ON. 2O(1010) ;turns ON outputs 1 & 3 and turns OFF 2 & 4. All ;others outputs remain unchanged.			
Note	When an output is set ON, the corresponding input bit is not available for use. Thus, outputs 5 to 8 should not be used (left OFF) if limits, registration or home are being used.			

OFF Shutdown motor power

Syntax aOFF	Units -	Range of 'n' -	Default OFF	See also ON, RFS
Description	Issuing an OFF command de-energises the drive to shutdown the motor power. The indexer 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.

The command allows execution of moves provided the motor is not on a limit.

Configuration system variables, such as MR, MC, MS are loaded into the drive's indexer when the ON command is issued. The current state of the drive fault registers will be cleared.

ON will clear the User Fault and Drive Fault variable to all zeros.

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

Example

1OFF	;drive must be OFF for configuration
1W(MC,50)	;configure motor current
1W(MR,400)	;configure motor resolution
1ON	;turn drive ON again

Note Issuing an ON command will clear all user status flags.

PROFILE Define move profile

Syntax	aPROFILEnumber(AA,AD,D,V,VS)
--------	------------------------------

Description The **PROFILE** command sets up a table of move profiles in the indexer 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
VStartStop	VS

Ranges for the **AA**, **AD**, **D**, **V** and **VS** 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 VS0.00*

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 rps² start stop speed of 1rps:

3PROFILE1(200,200,1500,25,1)

Profile 2 is to represent a move of 4800 steps on axis 3 at a velocity of 45 rps, acceleration of 150 rps² deceleration of 200 rps², start stop speed of 1rps:

3PROFILE2(150,200,4800,45,1)

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
Start/Stop velocity	1	1

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	<pre>0PS ;global pause 1D4000 ;setup axis 1 1V5 ; . 1A50 ; . 2D8000 ;setup axis 2 2V10 ; . 2A100 ; . 0G ;global GO 0C ;global continue</pre>			
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 . Refer to the table of system parameters for more information within Control of XLi Drives or later in this section.			
Properties	Immediate or buffered, can be used in labelled block, not saved by SV Note: 1R(RB) is immediate only			
Example	<pre>2W(MR,4000) ;set variable MR (motor resolution) to be 4000 2R(MR) ;report the current value of variable MR The response will be *4000</pre>			

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 indexer 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 (0 to 8). The default value is 0 (no output). The user program must reset this output before the next move.

For REG to work, the indexer must be globally armed (ARM1). See **ARM**.

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	
Registration window	0 to 2147483647	steps	
Output	0 or 8		Default is 0

Example**2START:****2PROFILE1(10,10,40000,5,1)****2PROFILE2(20,20,20000,10,5)****2REG1(1,1,10000)****2USE(2)****2G****2END****2REG:****2O(XXXXXXX1)** ; Turn output 8 on : increment batch counter**2T0.5** ; Delay execution for 500mS**2O(XXXXXXX0)** ; Turn off output 8**2END****2NOREG:** ; if we come here we didn't have a valid reg mark**2O(XXXXXXX1X)** ; Turn op7 on :push unlabelled product off conveyor**2T0.25** ; Delay execution for 250mS**2O(XXXXXXX0X)** ; Turn off output 7**2END**

RFS

Return to factory settings

Syntax aRFS	Units -	Range of 'n' -	Default -	See also SV
----------------	------------	-------------------	--------------	----------------

Description Issuing an **RFS** command initialises the indexer 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
 A & AD=10 rps²
 ARM*, HOME, LSEL, REG all disabled
 CQ=1 (queuing enabled)
 D=+4000
 EX=3 (RS232)
 IP=1 (in position)
 IT=10
 LIMITS enabled
 MC=50
 MI (Mode incremental)
 MR=4000
 MS=50
 MV=0 (not moving)
 PA=0
 PI=0
 SN (scan time) set to 50 milliseconds
 V=1 rps
 VS=0 rps

Note: the RFS state of ARM is disabled, but the fault label is 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 KILL

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.

The command may be used either 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

SN Scan command

Syntax	Units	Range of 'n'	Default	See also
aSNn	milliseconds	1 to 1000	50ms	LSEL

Description The **SN** command allows you to set an input scan-rate time in milliseconds. This is the rate at which the inputs are read. For use with a plc, the scan time should be set to a time less than the on-time of the plc outputs.

The default time applies to all inputs from power-up.

Setting SN too short may make the drive vulnerable to electrical pulse noise on the drive's user input lines.

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

Example **1SN10** ;set scan time to 10ms
 1SN ;report value of scan time
 **10*

SV Save configuration

Syntax aSV	Units -	Range of 'n' -	Default -	See also -
Description	<p>When the SV command is issued, the current indexer system variables and user programs are stored in non volatile memory. Any data saved, will be restored following the next power-ON cycle.</p> <p>This command will not operate whilst the motor is being commanded to move or whilst executing a program.</p> <p>Wait 1-2 seconds before sending any other command following an SV.</p>			
Properties	Immediate or buffered, can't be used in labelled block			
Example	<p>1RFS ;return drive to factory settings</p> <p>1A150.1 ;acceleration set to 150.1rps²</p> <p>1SV ;save current settings</p> <p>1Z ;reset drive</p> <p>1A ;report current value of acceleration</p> <p><i>*150.1</i></p>			

T Time delay

Syntax aTn	Units seconds	Range of 'n' 0.05 to 10	Default none	See also IF
Description	<p>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 (<i>*E out of range</i>).</p>			
Properties	Immediate or buffered, can be used in labelled block, not saved by SV			
Example	<p>4T6 ; delay for 6 seconds</p> <p>4T0.38 ; delay 0.35 seconds (rounded down)</p>			

TR

Wait for trigger

Syntax	aTR(system_variable,relation,value)								
Description	<p>The TR command pauses command execution until the trigger condition is met.</p> <p>The trigger condition is met if the relation between system_variable and value is true.</p> <p>Valid relations for the comparison are:</p> <table><tr><td>=</td><td>Equals</td></tr><tr><td><></td><td>Does not equal</td></tr><tr><td>></td><td>Greater than</td></tr><tr><td><</td><td>Less than</td></tr></table> <p>Value is a number generated by the system_variable being tested. Refer to the system variables table for more information.</p> <p>Refer to the table of system variables that can be used for conditional control.</p>	=	Equals	<>	Does not equal	>	Greater than	<	Less than
=	Equals								
<>	Does not equal								
>	Greater than								
<	Less than								
Properties	Immediate or buffered, can be used in labelled block, not saved by SV								
Example	<p>3TR(PA,>,2000) ; wait for position absolute to be >2000 steps</p> <p>3TR(IN,=,XXX11XXX) ; wait for user inputs 4 and 5 to be high</p>								
Notes	<p>If you wish to use the TR command during motion, command queuing (system variable CQ) must be set for continuous execution (CQ=0).</p> <p>Issuing a K or S from the command line will clear a trigger condition.</p> <p>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.</p>								

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, saved by SV			
Example	1PROFILE1(200,20,1500,25,1) ;define profile 1 1PROFILE2(150,200,4800,45,1) ;define profile 2 1USE(2) ;use motion profile 2			

V Velocity

Syntax	Units	Range of 'n'	Default	See also
aVn	rev/s	0.01 to 50.00	1	VS
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	<p>[1] The maximum stepping rate is limited to 200,000 steps per second. With a maximum drive resolution of 4000 steps per second the maximum velocity will always be 50.0 rev/s.</p> <p>[2] Programmed velocities less than the start-stop velocity VS will result in a constant-velocity move with no acceleration or deceleration.</p>			

Note [3] In a positioning move the deceleration time is limited to 65 seconds. If a very low value for deceleration is used together with a high peak velocity such that the deceleration time is greater than 65 seconds, the drive will report an error *E when the G command is given.

[4] 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.

VS Start stop velocity threshold

Syntax	Units	Range of 'n'	Default	See also
aVSn	rev/s	0.00 to 5.00	0	V
Description	The VS command sets or reports the velocity threshold value above which the motor starts to accelerate towards its final velocity set by V, or below which the motor will start and stop with no acceleration/deceleration.			
	Use of the VS command can shorten move times and avoid low speed resonance. VS must be set to a value within the pull-in speed of the motor. Note: Velocity V can be set below this threshold value for moves made at sub-threshold velocities. All moves made at a velocity less than VS will be made without any acceleration or deceleration time.			
Properties	Immediate or buffered, can be used in labelled block, saved by SV			
Example	To set the start-stop velocity threshold of axis 5 to 0.5 rps, type 5VS0.5			
	To report the current start-stop velocity threshold of axis 5, type 5VS			
	For a start-stop velocity threshold of 0.5 rps on axis 5, the controller responds with *0.50			
	No units are reported.			

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 in **Control of XLi Drives** for more information.

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

Example Set system variable MR (drive resolution) to 4000 **2W(MR,4000)**
Report the current value of system variable MR **2R(MR)**
The controller responds with..... *4000

Note The drive will need to be de-energised before certain system variables such as MR, MC or MS can be written to.

See also R command.

Z Reset

Syntax	Units	Range of 'n'	Default	See also
aZ	-	-	-	-

Description The **Z** command resets the drive's indexer. This is similar to power cycling the indexer. Upon restart, the user program following the START: label will execute only if the ARM command = 1.

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

Example To reset all drives, type..... **0Z**

Set comms address remotely

Syntax	Units	Range of 'n'	Default	See also
a#n	-	1 to 255	1	-

Description This command (#) allows you to set the unit address via software. Upon receipt of the command, the indexer will send a #n+1 command along the daisy chain provided the echo mode is set.
To address a specific axis place the current address 'a' before the # symbol.

Following an SV (save) command, the address will be stored in non-volatile memory and will be used until the indexer is re-addressed.

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.

“ “

Quote command

Syntax	Units	Range of 'n'	Default	See also
a“ “	-	-	-	-
Description	Commands can be sent to other drives or messages sent to remote displays using QUOTE. Using the RS232 link, up to 10 ASCII characters can be used 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	<pre> 1DECLARE(EXAMP) ;declare label 1EXAMP: ;label EXAMP 1"*TEST" ;quote TEST 1END ;end label 1GOTO(EXAMP) ;goto label EXAMP *TEST ;output message </pre>			
Note	To speed-up communications when addressing a number of drives, precede the quoted text with an asterisk '*'. Axes, apart from the one being addressed, will ignore the quoted text and this will save processing time.			

System Variables

Var	Name	R	W	Range/default value
BU	Buffer usage	Y	N	% of program buffer used 0-100
CQ	Command queuing	Y	Y	1= Pauses until move complete (default) 0= continuous execution
DF	Drive Fault status	Y	N	See following section
EI	Encoder Input	Y	Y	N/A
EO	Encoder signal output	Y	Y	N/A
EP	Encoder Position	Y	Y	N/A
ER	Feedback encoder resolution	Y	Y	N/A
EX	Comms. Response Style & echo control	Y	Y	0= speak when spoken to, echo off, RS485,multi axis 1= speak whenever, echo off, RS485, single axis 2= speak when spoken to, echo on, RS232,multi axis 3= speak whenever, echo on, RS232, single axis
IN	Inputs	N	N	Same format as IS command
IP	In Position flag	Y	N	1= In position or 0= not yet in position
IT	In Position Time	Y	Y	1-500mS, default=10mS
MC	Motor current	Y	Y	50, 60, 70, 80, 90, 100 (in percent), default see RFS
MR	Motor resolution	Y	Y	400, 800, 2000, 4000, default see RFS
MS	Motor standby current	Y	Y	50, 70 (in percent) , default see RFS
MV	Moving	Y	N	Flag 1= moving or 0 = not moving
PA	Position Absolute	Y	Y	-2,147,483,648 to 2,147,483,647 steps/counts, default 0
PE	Position Error	Y	N	motor steps
PI	Position Incremental	Y	Y	-2,147,483,648 to 2,147,483,647 steps/counts, default 0
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
SN	Serial number	Y	N	Drive serial number
ST	Status of indexing	Y	N	See Reporting the Status of Variables
UF	User program Fault status	Y	N	See Reporting the Status of Variables

Drive Faults

Bit Number	DF Information
1	Composite fault
2	Output stage over current
3	Supply rail failure
4	Ambient over temperature
5	Drive over temperature
6	Configuration error
7	Motor high voltage rail failure
8	Output fault
9 to 32	Reserved

Status Bits

Bit Number	Status Information
1	Command processing paused
2	Looping (command executing)
3	Wait for trigger (input)
4	Running program
5	Going home
6	Waiting for delay timeout
7	Registration in progress
9	Motor energised
11	Event triggered - active until trigger inputs are reset
12	Input in LSEL not matching label
13	-ve limit seen during last move
14	+ve limit seen during last move
19	Moving (In Motion)
20	Stationary (in position)
21	No registration signal seen in registration window
22	Cannot stop within the defined registration distance
23 to 32	Reserved

User Faults

Bit Number	UF Information
1	Value is out of range
2	Incorrect command syntax
3	Last label already in use
4	Label of this name not defined
5	Missing Z pulse when homing
6	Homing failed - no signal detected
7	Home signal too narrow
8	Drive de-energised
9	Cannot relate END statement to a label
10	Program memory buffer full
11	No more motion profiles available
12	No more sequence labels available
13	End of travel limit hit
14	Still moving
15	Deceleration error
16	Transmit buffer overflow
17	User program nesting overflow
18	Cannot use an undefined profile
19	Drive not ready
22	Save error
23	Command not supported by this product
24 to 32	Reserved

Command List

Command	Description
A	Acceleration/Deceleration
AA	Acceleration
AD	Deceleration
ARM	Enable event triggered code
C	Continue
CLEAR	Clear user code
DECLARE	Declare
D	Distance
E	Enable/disable communications
EXIT	Exit from loop
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
LOOP	Repeat user code
LSEL	Label select
M	Mode
O	Output
OFF	Shutdown motor power
ON	Turn on motor power

Command	Description
PROFILE	Define move profile
PS	Pause
R	Report system parameter
REG	Registration move
RFS	Return to factory settings
S	Stop
SN	Scan command
SV	Save configuration
T	Time delay
TR	Wait for trigger
USE	Use
V	Velocity
VS	Start stop velocity threshold
W	Write system variable
Z	Reset
#	Set comms address remotely
“ ”	Quote command

7. XLi Maintenance and Troubleshooting

Maintenance

XLi 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 - Danger of electric shock

High voltages are present within the drive module and on the motor phase outputs. Do not dismantle the drive cover, there are no user-serviceable parts inside.

Factory Settings

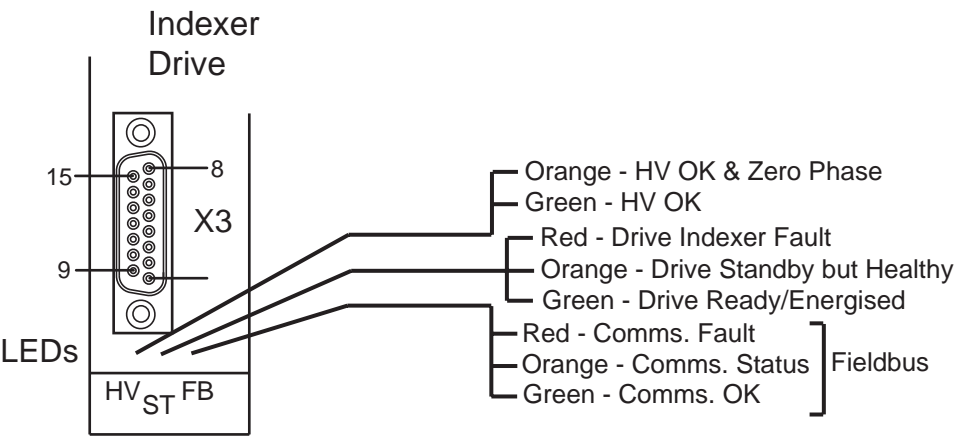
To return a drive to its factory settings, use **Direct Mode** and type in the following commands:

1OFF	; turn OFF before performing RFS
1RFS	; return to factory settings
1SV	; save the factory settings

Drive LED indications

In the event of any system malfunction, the first thing to check is the status of the LEDs on the front of the drive module. If no LEDs are illuminated, verify that power is turned on and that the nominal input DC voltage is present at the X1 connector. If it is, but there are still no LEDs illuminated, the drive is faulty and needs to be returned for repair.

The drive LEDs are 3-colour devices and provide information on the state of the drive as shown in Figure 7-1.



2. The power input voltage is either too high or too low. Check that the supply is switched ON and is within the limits specified in the Hardware Reference Section.
3. The drive has shut down due to excess temperature rise. Check that the ventilation slots at the top and bottom of the housing are not obstructed and that there is free flow of air through the drive. Make sure that the drive is not mounted directly above a source of heat such as a power supply, and if possible check the ambient temperature in the vicinity of the drive casing. Allow the drive to cool down and re-check. If the motor has ample torque for the application, try reducing the current setting which will reduce the drive dissipation.
4. An external 24V DC supply is used and has been connected the wrong way round. Check the polarity of the external supply - positive should be on X1 pin 4 and negative (or 0V) on X1 pin 5.
5. There is an internal fault in the drive or with the internally-generated logic supply. In either case the drive will need to be returned for repair.

Further information on drive fault analysis will be found later in this section.

Right LED

This LED has no significance unless the optional Fieldbus module has been fitted.

Drive fault analysis

Drive or system problems indicated by the centre drive LED showing red may be analysed for further information as outlined below. The red LED indicates that a fault has caused the drive output stage to be turned OFF (de-energised). Once the fault has been corrected the drive may be re-energised using the ON command (unless the fault is the result of an internal failure).

Drive faults can be identified by performing a read of the system variable DF (Drive Fault) using the R report system parameter command. A list of drive faults is given in Section 5, but is also reproduced in Table 7-1 below:

Use the report command to display the DF word pattern, that is '**aR(DF)**', or use EASI-TOOLS Drive status report to help diagnose drive problems.

Bit Number	DF Information
1	Composite fault
2	Output stage over current
3	Supply rail failure
4	Reserved
5	Drive over temperature
6	Configuration error
7	Motor high voltage rail failure
8	Output fault
9 to 32	Reserved

Table 7-1. Drive Fault Bit Description

Drive Fault Details

Composite Fault

A composite fault is an internal drive fault which may occur in combination with another fault listed below. If it is the only fault reported and it is not cleared by cycling the power, the drive is faulty and should be returned for repair.

Output Stage Over Current Fault

This form of fault can be caused by a short-circuit between motor phases or a motor phase to earth short.

Supply Rail Failure

This fault can be caused by one of the following conditions:

- Drive supply missing
- HT out of specification
- +24V supply out of specification
- Internal drive service supply out of specification

Drive Over Temperature

Ambient temperature inside the case of the drive is too hot.

Configuration Error

Internal drive configuration error has occurred or configuration has been interrupted at start-up.

Motor High Voltage Rail Failure

This fault indicates that the HV output is out of specification.

Output Fault

This fault is caused by the user outputs drive chip suffering a current overload on one or more of its outputs or suffering a thermal shutdown.

Further checks

If you can safely do so, check that there is holding torque from the motor. However, take care not to touch any part of the system if this could result in personal injury should unexpected motion occur - see the safety warning at the front of this User Guide.

Drive LEDs OK but no holding torque from the motor

Disconnect power, remove the motor connector X1 and check the motor connections for continuity across each phase. Check that the drive has not been shut down by an OFF command as detailed above.

Motor has holding torque but will not run

The most likely cause is lack of communication between the PC and the drive. This can be checked by attempting a Connect from the Terminal menu of EASI-TOOLS. If the connection fails with the following screen displayed:

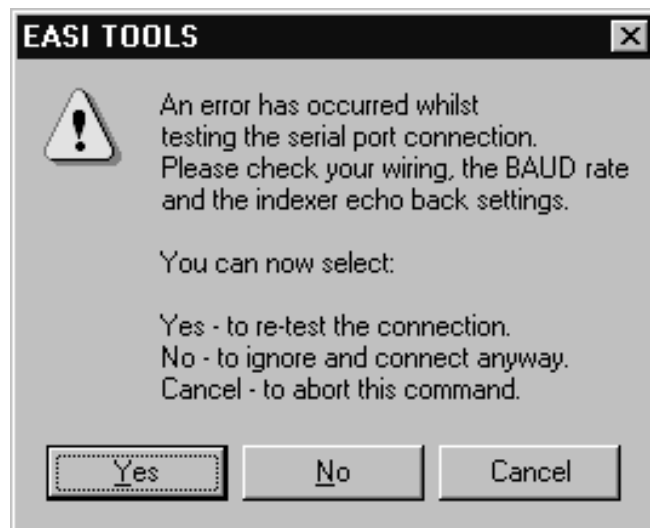


Figure 7-2. Communications Failure Error Message

Check the following:

Ensure that the serial port configuration is set correctly in EASI-TOOLS and the correct serial COM port has been selected. If you experience PC IRQ problems, read the note in the software installation section to allow re-allocation of ports.

Check that the drive address setting is correct - see the Installation Section.

If none of the above resolves the lack of communications, 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 lines 4 and 5 on the cable. In this condition any command sent from the terminal window should be echoed back, confirming the integrity of the overall RS232 link. If this works, but you cannot communicate with the drive, contact our Technical Support Department or your ATC for further advice.

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 - XL25i, XL50i & XL80i

Functional Specification

Parameter	Value
Amplifier type	MOSFET chopper
Motor resolution	400, 800, 2000, 4000 steps/rev
Maximum stepping rate	200kHz at 4000 steps/rev
Nominal chopping frequency	28kHz (XL25i), 20kHz (XL50i & XL80i)
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
Maximum output current	XL25i: 2.5A peak +/-10% XL50i: 5.0A peak +/-10% XL80i: 8.0A peak +/-10%
Output current adjustment	50% to 100% of maximum current, software selectable in 10% increments
Standby current reduction	50% or 70% selected by software
Standby reduction time	30mS from last step pulse
Nominal motor bus voltage	80V DC +5%
Supply voltage, DC operation	XL25i: 24-80V +5% XL50i: 48-80V +5% XL80i: 56-80V +5%
Indexer logic supply input	24V DC +10% -15%
Logic supply current	800mA max. (all outputs ON at max. current)
LED status indicators (tri-colour)	HV OK/zero phase, drive OK/energised/fault, comms OK/fault
Motor inductance range	0.5 - 10mH (0.8 - 10mH preferred range)

Table 8-1. Functional Specification

Indexer Specification

Parameter	Value
Command Interface	
Position range	+/- 2,147,483,647 steps
Velocity range	0.01 to 50 revs/sec
Acceleration/deceleration range	0.1 to 1024.0 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
Address setting range	1 -255 by software
RS232 connection	2 wire plus ground
Digital Inputs	
User programmable inputs, shared with user outputs	8 (home & limits can be used as additional user-programmable inputs but with limited functionality)
Dedicated inputs	Home, + limit, - limit, registration
Input levels	Logic high 13 - 30V, logic low 0 - 3.3V
Input impedance	2.3K to 0V
Digital Outputs shared with inputs (cannot use input if output is used)	
User-programmable outputs	8, current sourcing outputs
Output levels	0 - 0.4V (output OFF with 2k load) 22 - 24V ±10% (output ON)
Output current rating	60mA maximum per output
Fanout of each output	4

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
Storage temperature range	-20 to 70°C
Humidity	5 to 95% non-condensing
Cooling	Natural convection
Housing	Aluminium
Protection class	IP20
Weight	0.4kg

* See Mechanical Installation Section

Table 8-4. Drive Environment Specification

XL_PSU Specification

XL_PSU Electrical Specification

Parameter	Value
Input voltage	
Nominal	110 to 230 V AC RMS
Min/Max	95 to 264V AC RMS
Mains supply frequency range	50/60Hz +/- 2Hz
Input power factor	Better than 0.9 at 250W load
Output voltage	80V DC $\pm 5\%$ (no load)
Output current	
230V AC input, no 24V load	3.1A continuous, 7.5A peak
230V AC input, max 24V load	2.6A continuous, 7.0A peak
<150V AC input, no 24V load	2.5A continuous, 6.2A peak
<150V AC input, max 24V load	2.0A continuous, 5.7A peak
Protection	Short circuit / overload protected
On board capacitance	6600 μ F
24V Auxiliary output	
Voltage	24V DC +10% -15%
Current	1.8A Max
Protection	Short circuit / overload protected
Output cable length restriction	not to exceed 10m in length
Total output power	250W continuous
Peak	600W for 1 second. Below input voltages of 150V AC, output power is 200W continuous, 500W peak.
Protection circuits	
XL_PSU Module	Overtemperature
Output	
Current	Short circuit / Overload
Power	Excessive continuous power
+24V supply	
Current	Short circuit
Voltage	Reverse polarity
Connection / disconnection	Hot plugging protection

Table 8-5. XL_PSU Electrical Specification

XL_PSU Mechanical Specification

Parameter	Value
Housing	Two part brushed aluminium case
Dimensions	
Depth	130 mm Max
Width	50 mm Max
Height	175 mm Max (195 mm at flange mount)
Weight	1 kg

Table 8-6. XL_PSU Mechanical Specification

XL_PSU Environment Specification

Parameter	Value
Pollution	Degree 2
Installation category	II*
Operating temperature range	0 to 50°C ambient
Storage temperature range	-20° to 70°C
Humidity	5 to 95% non condensing
Altitude	2000 metres
Ingress protection	IP20
Cooling	Natural convection with integral fan. Unit to be mounted vertically with at least 50mm free space above and below the package. Side clearance of 10mm free space both sides.

Table 8-7. XL_PSU Environment Specification

*Note: Installation category (also called Overvoltage Category) specifies the level of mains voltage surges that the equipment will be subjected to. The category depends upon the location of the equipment, and on any external surge protection provided. Equipment in an industrial environment, directly connected to major feeders/short branch circuits, is subjected to Installation Category III. If this is the case, a reduction to Installation Category II is required. This can be achieved by use of an isolating transformer with an earthed screen between primary and secondary, or by fitting varistors from live to neutral and from neutral to earth. Varistors for a 120V AC input should be rated to absorb at least 40 Joules. For example, DIN-rail mounted surge suppressor RS 240-062 (110V AC, 70J) or Farnell 294-275 (110V AC, 80J) would be suitable. Varistors for a 240V AC input should be rated to absorb at least 80 Joules. For example, DIN-rail mounted surge suppressor RS 240-078 (240V AC, 140J) or Farnell 294-287 (240V AC, 140J) would be suitable.

Index

#

set address remotely, 132

+

+24V fuse rating, 19

+24V supply connections, 19

+24V supply lead length restrictions, 19

A

A acceleration deceleration, 93

A indexer mode, 118

AA acceleration, 93

Absolute preset moves, 59

Absorber ferrite, 24

AC supply volts, 10

AD deceleration, 94

Address

incrementing, 39

RS232 setting, 39

saving, 39

setting sequential patterns, 39

software maximum, 39

software selection, 39

software selection limitations, 39

using the hash command, 39

Arm command, 46

ARM command, 49

ARM event code, 95

B

Baud rate selection, 80

Bipolar rating, 24

Braking resistor, 16

Braking resistor calculations, 14

Braking resistor connections, 15

Bus voltage, 145

C

C continue, 97

Cabinet installation, 9

Cable screening, 22

CAN variants, 2

Chop frequency, 145

Clean earth, 9

CLEAR user code, 97

Clockwise motion, 59

Code

structure, 42

Command

address requirements, 89

ARM, 49

label & multi parameter syntax, 90

presentation, 89

reference, 89

simple syntax, 89

Command checking, 92

Command checking examples, 92

Command defaults, 125

Command list, 137

Command properties, 90

Command queuing, 58

Commands sent

waiting on a trigger, 56

Communications specification, 146

Composite fault, 142

Conditional code, 57

Configuration button settings, 82

Configuration fault, 142

Confirming drive operation, 87

Continuous moves, 59

Control signal wiring, 24

Cooling, 147

Cooling requirements, 3

D

D distance, 99

DC supply volts, 10

Declarations

position within the code, 42

Declare

command, 42

examples, 42

DECLARE, 98

Dedicated I/O, 32

Default command and system variable
settings, 125

Default directory, 76

Device addressing, 44

DF bit order, 54

DF word, 54

Digital inputs specification, 146

Digital outputs specification, 146

Dimensions, 4

Direct mode, 41
Disconnecting device, 9
Drive
 cooling, 3
 fault reporting, 54, 142
Drive connector layout, 30
Drive fault
 byte reporting, 54
Drive faults, 53
Drive faults list, 135
Drive inspection, 139
Drive over temperature fault, 142
Dump resistor connections, 15
E
E enable/disable comms, 100
Earth arrangements, 9
EASI-TOOLS
 axis configuration, 82
 change axis address, 81
 communicating with drive, 80
 current configuration, 83
 downloading, 86
 drive configuration, 81
 encoder configuration, 84
 help, 86
 installation directory, 77
 limits configuration, 84
 menu overview, 78
 other configuration, 84
 prg files, 80
 resolution configuration, 83
 running, 77
 startup, 78
 status reporting, 85
 uninstall, 77
 uploading, 86
Edit menu, 78
Electrostatic precautions, 9
EMC
 typical installation, 21
EMC filter
 spacing, 20
EMC Installation, 20
 general requirements, 20
EMC wiring
 recommendations, 20
END label, 42
Environment specification for drive, 147

Event
 trigger, 50
Event code example, 49
Event driven code, 49
EXIT from loop, 101
F
Factory settings, 139
Fanout of outputs, 35
FAULT, 46
Fault label, 46
 call conditions, 47
 conditions of execution, 46
 example, 48
 non call conditions, 47
 table, 47
Fault output, 35
Fault status reporting, 53
Ferrite absorber
 Curie temperature, 24
 impedance, 24
 part number, 24
 size, 24
Ferrite absorber specification, 24
File menu, 78
Function indicators, 2
Fuse for +24V supply, 19
G
GH go home, 103
GO, 102
Go home command, 68
Go home while in the home switch, 69
GOSUB go to subroutine, 103
Goto line number box, 79
GOTO routine, 105
H
H change direction, 106
Help menu, 79
Home
 approach speed, 68
 configuration, 66
 direction of travel, 70
 mode 0, 67
 mode 1, 67
 mode 2, 67
 mode selection, 67
 modes, 67
 switch considerations, 66

HOME, 108
 definition of terms, 65
Home switch too narrow, 68
Homing
 operations, 65
Housing material, 147
Humidity, 147

I

I indexer mode, 118
I/O lead length restrictions, 31
IF command, 57
IF test, 110
Immediate commands, 91
Immediate or buffered commands, 91
Immediate properties, 90
Incremental preset moves, 59
Indexer specification, 146
Input events, 49
Input response time, 33
Inputs & outputs, 2
Installation, 9
Installation Category, 3
Installation safety requirements, 9
Installation setup program, 76
Installation time for S/W, 76
Interrogation commands, 56
IRQ problems, 80
IS input status, 111

K

KILL, 112

L

Label
 definition, 42
 execution, 43
 naming, 44
 number available, 42
 predefined list, 42
 select command, 43
 specification, 44
 system, 46
Labelled block properties, 91
Large motors, 26
LIMIT changes for stop, 73
Limit inputs, 32
Limit switch wiring, 73
Limit switches, 73
LIMITS, 113

Line fuse type, 19
Line fuse values, 19
LIST user program, 114
Logic input circuit, 33
Logic input circuit arrangements, 33
Logic inputs, 33
Logic output circuit, 34
Logic outputs, 34
Loop command, 45
LOOP user code, 115
Low voltage operation, 18
LSEL command, 43
LSEL example, 49
LVD requirements, 9

M

Maintenance, 139
Mode command, 59
Monitor type, 75
Motion
 profiles, 60
Motor
 cable lengths, 22
 connections, 22, 26
 default settings, 26, 27
 inductance values, 24
 largest size, 26
 minimum inductance values, 24
 safety earth connection, 27
 selection, 24
 system variables, 26
 wire size, 22
Motor connector X1, 36
Motor direction note, 60
Motor high voltage rail fault, 142
Motor inductance check, 75
Motor inspection, 139
Motor mounting precautions, 7
Motors
 4 lead, 24
 6 lead, 24
 8 lead, 24
 voltage rating, 26
Mouse, 75
Move
 types, 59
Moves
 absolute preset, 59
 continuous, 59

- incremental preset, 59
- preset, 59

N

- NOREG, 46
- NOREG label, 63

O

- O output, 119
- OFF shutdown motor, 119
- ON turn on motor, 120
- Output connections, 12
- Output current capacity, 34
- Output fault, 142
- Output relay drive circuit, 34
- Output stage over current fault, 142
- Overload of outputs, 27

P

- P1 connector part number, 12
- P2 connector part number, 12
- PA system variable, 66
- Parallel connections, 26
- Parallel motor wiring connections, 29
- Parameter checking, 92
- Parameter value checking, 92
- PC requirements, 75
- Point to point positioning, 62
- Pollution degree, 147
- Port configuration, 80
- Position flags, 52
- Position time (IT), 52
- Power input cable size, 17
- Power wiring precautions, 17
- Preset moves, 59
- Product
 - description, 1
 - features, 2
 - types, 1
- Product features, 2
- PROFILE of a move, 120
- Program
 - examples, 43
- Program size, 75
- Program structure, 43
- Programmed mode, 41
- Properties
 - immediate, 91
 - labelled block, 91
 - save, 92

- Properties of commands, 90
- Protection circuits, 2
- Protection class, 147
- PS pause, 122
- PSU connecting links, 13
- PSU discrete design, 17

Q

- Quote command, 133

R

- R report system parameter, 122
- REG, 46
- REG label, 63
- REG registration move, 123
- Registration, 63
 - problems, 63
- Registration example, 64
- Registration output, 63
- Report commands that can be saved, 56
- Returning the system, 144
- RFS return to factory settings, 125
- RS232
 - cable requirements, 37
- RS232 connections, 37
- RS232 connector, 37
- RS232 leads, 38

S

- S stop, 126
- Save properties, 92
- Screened I/O connections, 9
- Search menu, 78
- Serial cables, 37
- Serial comms connector, 37
- Serial communications configuration, 80
- Serial link 25 way connector, 75
- Serial link lead, 75
- Series connections, 26
- Series motor wiring connections, 28
- Setup file, 76
- Short circuit protection, 27
- SN scan, 126
- Software file size, 76
- Software installation, 76
- Software requirements, 75
- ST bit order, 52
- ST motors
 - optimum types, 25
- ST word, 52

Star point, 9
 START, 46
 START label, 49
 Starting a program, 41
 Status Bit description, 53
 Status bits list, 135
 Status reporting
 immediate, 56
 Status variable
 byte reporting, 53
 Status variable reporting, 52
 Stop input, 73
 Supply +24V, 19
 Supply connections, 12
 Supply rail fault, 142
 Supply volts, 10
 SV save configuration, 127
 SY motors
 optimum types, 25
 Syntax checking, 92
 System labels, 46
 System variables, 50, 89
 reading, 50
 reporting status, 52
 table of, 51
 testing, 50
 writing, 50
 System variables default settings, 125
 System variables list, 134

T

T time delay, 127
 Temperature
 ambient, 147
 storage, 147
 Terminal menu, 79
 Test code, 87
 TR command, 57
 TR wait for trigger, 128
 Transformer
 multi axis systems, 18
 sizing for applications, 18
 VA rating, 18
 Transformer selection guide, 18
 Trapezoidal profile, 61
 Trapezoidal profile modified, 62
 Triangular profile, 60

U

UF byte, 55
 USE, 129
 USE command, 50
 User fault
 byte reporting, 55
 clear conditions, 56
 test example, 56
 User fault descriptions, 55
 User fault reporting, 55
 User faults, 53
 User faults list, 136
 User I/O configuration, 33, 34
 User outputs, 34
 Utilities menu, 79

V

V velocity, 129
 Virtual inputs, 32
 VS start stop velocity threshold, 130

W

W write system variable, 131
 Weight, 147
 Welcome box, 76
 Windows menu, 79
 Windows™, 75
 Wire size of motor earth, 27
 Withstanding voltage rating, 17

X

XL_PSU
 dimensions, 149
 electrical specification, 148
 environment specification, 150
 input voltage, 148
 LED indicators, 16
 mechanical specification, 149
 output voltage, 148
 power output, 148
 product description, 11
 weight, 149
 XL_PSU drive wiring diagram, 13
 XL_PSU mounting information, 6
 XL_PSU product appearance, 11
 XL25i
 supply volts, 10
 XL25i/XL50i drive
 dimensions, 4

XL50i

supply volts, 10

XL80i

supply volts, 10

XL-connect kit, 13

ZZ reset, 131

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

