ACS Tech80

REDEFINING MOTION CONTROL

SB214 Series

Multi-axis Motion Controllers

Hardware & Setup Guide

Version 1.01 (August 1996)

Part number: MN-00214-000

ACS Tech80

http://www.acs-tech80.com

North America:
658 Mendelssohn Avenue North,
Minneapolis, MN, 55427, USA
Tel: +1 763-542-9545
Fax: +1 763-542-9785
E-mail: info@acs-tech80.com
support@acs-tech80.com

Rest of the World:
Ramat Gabriel Industrial Park,
POB 5668, Migdal Ha'Emek, 10500, Israel
Tel: +972 6-6546440
Fax: +972 6-6546443
E-mail: info@acs-tech80.co.il
support@acs-tech80.co.il

Warning

Dangerous voltages are present in this equipment!

Contact with live parts could cause serious injury or death!

Refer connection, installation, maintenance, adjustment, servicing and operation to qualified personnel.

Table Of Contents

Table Of Figures

1. OVERVIEW

The SB214 is a state of the art line of multi axis motion controllers. It is based on a proprietary 40MHz Servo Processor ® that has been specifically developed to meet the demanding needs of high performance machinery and automated systems. Each axis is controlled by its own dedicated Servo Processor. This enormous processing power, combined with advanced programming techniques, provide unprecedented level of performance. 20kHz sampling rate, independent on the number of axes, flexibility and simple operation.

Four versions are available:

SB214PC	PC based four axis (and three axis) controller. An optional distribution box that simplifies the interface is available - SB214INT.
SB214PC2	PC based dual axis controller. An optional distribution box that simplifies the interface is available - SB214PC2INT
SB214VME	VME based four axis controller.
SB214ND	Stand alone, panel mounted four axis controller. This module combines the SB214PC, The SB214INT and all necessary power supplies onto a one rugged metal box.

All controllers share the same basic architecture, with the same software capabilities. Each one can control any type of motor - DC brush, DC brushless, AC induction and stepper, thus simplifying the task of mixing several motor types in an application.

Excellent Dynamic Range

Velocity range of up to 20,000,000 counts/second, and acceleration of 127,000,000 counts/sec² enables the use of very high resolution encoders and laser interferometer feedback devices without sacrificing high speed capabilities. Whether your performance needs are position accuracy, excellent velocity regulation, or very short settling times, a SB214 will deliver.

Advanced Control Algorithm

An advanced digital filter, sophisticated velocity observer, automatic velocity feed forward, acceleration feed forward, sampling rates of 20kHz, smoothed trapezoidal profiles, provide the ability to implement high bandwidth, responsive servo loops with exceptional dynamic tracking and settling performance.When using the ACS Adjuster® for Windows, a simple tuning algorithm tunes the filters for optimal, robust and stable system.
The velocity observer derives the velocity information from the encoder reading in an optimal way, thus providing excellent velocity control at low speeds. The observer also includes a second order low pass filter. If your

system suffers from a low frequency resonance - belt driven systems, and direct drive systems with high inertia are two common examples, you will find that the ability to program the bandwidth of that filter will allow you to handle those type of difficult mechanics with well tuned results.

Flexible Motion Control

The SB214 can be operated in many different motion modes: position, velocity, manual joystick, master/slave follower, CAM follower, registration move, arbitrary path, linear and circular interpolation, search for contact and index/homing searching.

The Master-Slave mode is characterized by its superb following accuracy, super-imposed move capability, ability to switch "on-the fly" from slave mode to velocity mode and vise-versa and by comprehensive software support. It has proven itself in the following applications: industrial flying shears, coil winding, multi-color paper and cloth printing, high accuracy laser scanning and plotting.

The Search for Contact motion mode is specially designed for Pick & Place applications, like wire bonders, die attach, SMD assembly and more. The Pick & Place head moves down in high speed and reaches a programmable point at a low search velocity. It continues to move down at that velocity until a contact or sensor is sensed. Afterwards it continues to move a programmable set distance beyond the contact position applying force. All exceptions - contact bouncing, contact not found and others are fully supported.

I/O Control

Uncommitted isolated logical I/Os and analog I/Os are available to the user. Interfacing with programmable logic controllers, thumbwheel switches, solenoids, relays, analog sensors and many other devices, is simple and straight forward.

Inputs are supported by automatic routines. You can define what process should be invoked when a specific input is activated. Once that input is activated, the process starts within 0.001 second.

Full Diagnostics

Built-in error routines, user defined automatic error handling routines (AUTO_ERR,AUTO_ES, AUTO_FLT, AUTO_ENX and more), hard wired limit and emergency switches, loss of feedback detection and soft travel limits protect the system against mistakes and system malfunctions. Every system failure; running into limit, exceeding the user defined error limit, drive failure, over current command, erroneous feedback etc. - is covered by an error message, giving the exact reason of failure. Recovery is fast and simple.

Powerful High Level Language

The powerful, yet simple, high level ACSPL programming language, enables the user to implement highly complex motion-time-event sequences and programs, with accurate positioning and timing. Automatic, user's programmable routines (AUTOEXEC, AUTO_IN# and more) eliminate the need for a host, and for start-up programming when the system is powered. Extensive editing capabilities, combined with on-line diagnostics, enable even the non-experienced user to implement his own applications in a matter of minutes. All programs can be stored in a non-volatile memory.

Advanced Features

The following advanced features are also supported:

- Real time data collection of one or two variables simultaneously, at a programmable rate of up to 1000Hz. Up to 1,024 samples can be captured per sampling. Built-in statistical functions (min., max., average) simplify the analysis of the collected data. The following variables can be captured: position, position error, Master position, short term velocity, command to drive (DAC output), analog inputs, output port, and input port.
- (Option) Real time Dual Axis Position Event Generator (PEG®). The state of seven outputs per axis can be altered, at a pre- defined, random set of 512 positions. In addition, a programmable pulse is generated at each point, or whenever passing a user defined position interval.

The combination of high level language, extensive command set, versatility, ease of operation, simple communication protocol and built-in protection schemes, enables the user to implement a very powerful control system, tailored to his exact needs, in minimum time, cost and effort.

Support Tools

The following programming support tools are available:

- **ACS Adjuster for Windows**.
  It provides a powerful, yet simple, tool for adjusting the digital drives. A built in S/W oscilloscope eliminates the need for an oscilloscope even when adjusting the high bandwidth current loops.
- **ACS C Library**
  A comprehensive **C library** for DOS, Windows (DLLs), Windows 95, and Windows NT.
- **ACS Debugger for Windows**
  The ACS Debugger is a Windows based debugging tool for ACSPL application programs. ACSPL commands and programs are executed by the controller in its application environment. Programs can be executed at full speed or one step at a time. Many breakpoints can be inserted. Execution hustory and actual values of program's variables can be monitored.

Documentation

The documentation consists of two guides:

- **ACS Software Guide**

It is a generic software reference guide common to all controllers and control modules that are produced by ACS.

It describes the programming language, the motion modes and special functions and includes a comprehensive reference section.

- **SB214 Hardware And Setup Guide**

It describes how to setup the module, how to interface with it, how to adjust the filters and specific motor setup.

This guide consists of the following sections:

Getting Started

This section describes how to set up the hardware for the various types of motors and drives, and explains how to communicate with a controller, execute moves, write ACSPL programs, edit, store, and load programs and parameters.

Digital Control Set up

This section describes how to set up the control algorithm, to operate properly an AC induction, DC brushless, DC brush and step motors.

Electrical Interface

This section describes how to connect the various functions of the controller with the outside world.

Communication

This section contains all the necessary information for communicating with the host computer or a terminal.

Input And Output Port

This section describes the digital and analog I/O ports.

Motion Monitoring

This section describes how to monitor real time motion variables.

2. SPECIFICATIONS

2.1 Architecture

CPUs

One, or two 20MHz Intel 80C196KD for high level tasks and managing.

Per axis - 40MHz SB2500 ACS Servo Processor for real time, high speed tasks.

Memories

EPROM: 256KX8 firmware.

RAM: 256Kx8 data storage. (SB214PC2 - 128Kx8)

Non-volatile Flash memory: 128KX8. 100,000 write cycles.

2.2 Control

Sampling rate:	20 kHz.
Position loop standard algorithm:	PI filter, acceleration feedforward, and automatic velocity feed forward. Anti-reset windup.
Velocity loop:	1’st order forward filter, 2’nd order velocity observer
Position Accuracy:	Within 1 count.
Range:	±999,999,999 counts.
Velocity accuracy:	
	Long term:0.005%.
	Short term: 0.01%-0.5% (system dependent).
Range:	1 to 20,000,000 counts/second. 1 count/sec resolution..
Acceleration Range:	1,000 - 127,000,000 counts/sec^2.
Drive command	two per axis. 10 bits PWM type or ±10V analog type. Opto-isolated.
Position feedback:	SB214PC/ND/VME (4 axis): X, Y - one encoder per axis Z,T - two encoders per axis SB214PC2 (2 axis): X - two encoders Y - one encoder maximum count rate: 20MHz.
Stepper pulse rate	1 to 2MHz pulse width: 0.4 to 60 microseconds.
Registration accuracy:	1 count at speed up to 5,000,000 counts/sec.
Index accuracy:	1 count at speed up to 5,000,000 counts/sec.

PEG® position compare accuracy:	1 count at speeds up to 100,000,000 counts/second.
PEG® Random mode repetition rate:	5 events per 0.001 second.
PEG® Incremental mode repetition rate:	up to 1MHz.
Serial Communication:	RS232 / RS422
Bus communication	8 bits data width via 1k deep fifo in each direction.
Uncommitted analog inputs:	SB214PC (4 axis): two, ±10V,10bits resolution. four, ±10V,12bits resolution. SB214VME (4 axis): two, ±10V,10bits resolution. SB214PC2 (2 axis): two, ±10V,10bits resolution.
Uncommitted analog outputs:	one general purpose, 8 bits resolution plus one per axis, 10bits resolution. All isolated.
Axis Dedicated Inputs:	Left & Right limits, Opto-isolated.
General Purpose Inputs:	16, opto-isolated.
General Purpose Outputs:	16, opto isolated.

Dimensions:

SB214PC	-	335x114[mm](WxH)
SB214PC2	-	244x114[mm](WxH)
SB214VME	-	160x233[mm](WxH)
SB214ND (box)	-	450x147x102[mm](WxHxD)

2.3 Main Features

- Powerful two to four axis universal motion controllers.
- DC brush, DC brushless, AC induction and step motor control.
- State-of-the-art proprietary Servo Processor Technology®.
- Fully digital control: velocity & position loops with anti windup at 20kHz sampling rate.
- Sinusoidal commutation of AC synchronous (=DC brushless) motors.
- AC induction motor control with servo level of performance.
- Automatic commutation setup of any 3 phase motor.
- Micro stepping control with 256/step resolution.
- Wide velocity range: 1 -20,000,000 counts/second.
- Easy programming with ACSPL high level language.
- Real time dual axis Position Event Generator (PEG®) at rates of up to 5kHz with ±1 count accuracy.
- "Teach & Go" of 1024 XYZT points.

- Built-in smart joystick interface.
- RS232 & RS422 serial communication interface.
- Fast bus communication via 1k deep fifo.
- Fully supported by ACS ADJUSTER for Windows.
- Master-Slave, Electronic Gearing and Cam operation.
- Advanced "Search for Contact" Motion Mode for pick & place applications.
- Programmable analog output for real time monitoring purposes.
- 16 inputs and 16 outputs.
- Powerful I/O handling with advanced PLC capabilities.
- Error Mapping accuracy enhancement.
- High speed registration mark input, with absolute accuracy hardware latching, supported by automatic routine and registration moves.
- Real time data collection (up to 1024 samples per second) and analysis.
- Comprehensive diagnostics and protection schemes supported by programmable automatic routines.
- A special high reliability serial communication protocol that is designed for systems that operate in an extremely noisy environment.
- Event driven mechanism that allows special events handling within 0.001 second.
- Automatic detection of encoder error and disconnection.

3. GETTING STARTED

This section describes how to set up the hardware for various types of drives and feedback, and explains how to communicate with a controller, execute moves, write ACSPL programs, edit, store, and load programs and parameters.

The complete details of the electrical interface are described in Section 5.

The following hardware is necessary:

1. One of the SB214XX Controllers.
1. Servo and/or stepper drivers.
1. Servo Motors (with encoders) or stepper motors.
1. PC with ACS Adjuster for Windows program.
1. (For serial communication) RS232, 3 wire cable, or RS422 cable.

All connections - RS232, motor, encoder, I/O, limits etc., must be done while the unit is disconnected from power.

3.1 SB214PC (4 axis)

3.1.1 Initial Setting

The SB214PC controller has factory initiated setting that may be changed according to your needs:

PC bus address:	100Hex.
Interrupt:	disabled.
Drive command type:	PWM (0-5V).
Isolation:	Full isolation between the drives and the controller.
Motor/Drive type:	Standard servo with single command.

See ELECTRICAL INTERFACE page 5-1 .

3.1.2 PC Bus Address

The SB214PC are factory set to communicate over the PC bus at I/O address 256 decimal (100 hex). The address is controlled by the setting of the dip switch SW8-SW2 (A9-A3). Unless there is a conflict at this address, it is recommended to use the default address. To change the address see Address Setup, page 5-18.

3.1.3 Interrupt

The cards are set at the factory to communicate with the PC without causing interrupts. The ACS Adjuster® for Windows does not use the interrupt capability and all interrupt setup jumpers must be left opened. To choose an interrupt see Interrupt Setup, page 5-18.

3.1.4 Drive Command Type

Two command types are supported: PWM (0-5V) type and analog type (± 10V). See Drive Interface, page 5-2.

3.1.5 Isolation

The card can be completely isolated from the drives (including the analog commands) and the digital I/O. To select whether you want an isolation from the Drivers, Inputs, Limits, Emergency Stop and/or Outputs see Isolation And External Supplies, page 5-9.

3.1.6 Using The Card Outside The PC (with serial communication).

To use the SB214PC as a stand-alone outside the PC, place JP31 and set DIP SW1 to "off" position. Do not forget to remove JP31 before inserting the board in a PC. See Serial Communication, page 5-9.

3.1.7 Installing The Board In A PC

The SB214PC can be installed in a 32, 16 and 8 bits PC bus.

Insert the SB214PC in a free slot. Make sure that there is enough space for the cables that are needed for the interface. You may leave the adjacent slot free and use its rear exit for the flat ribbon cables.

Fasten the board's rear plate with a screw.

3.1.8 Connecting To The SB214INT Distribution/Interface Box

Install the flat ribbon cables J5 (rear),J1, J2 and J4.

SB214INT	SB214PC
JA	J5
JB	J1
JC	J2
JD	J4

The **SB214INT** uses **5V** that is supplied via the **5V** pin. You may use the **5VPC** (by shorting it to the **5V** pin in the **ext. supply** connector) or use an external supply, if the **PC 5V** is not sufficient. The external supply must be connected between the **5V** and **RTN** pins.

The **SB214INT** is shipped with a short between the **5V** pin (which is the power source for the interface board) and the **5VPC** pin.

3.1.9 Powering Up The System

Power up the PC and the external supplies. The on board LED should blink for a short period.

Assuming that the ACS Adjuster® for Windows is already installed on your PC, activate it, set the address and switch to terminal mode.
While in terminal mode, press **<cr>**. The controller should respond with the **0>** prompt. Now type **RG1<cr>**. The controller should respond with a list of all the parameters of Group #1.

3.1.10 Motor Types

The motor type is set by the $XT (aXis Type) parameter. Change the type only when the axis is disabled. Afterwards execute a **SAVE** command and then a **HWRES** command.

XT=0	Stepper control with current commutation (with DR170 drivers made by ACS) - Two command signals per axis.
XT=1	Standard DC servo control with one command signal per axis.
XT=2	Stepper control. Pulse-Direction pair of signals per axis. Can be interfaced to standard stepper/micro-steppers drivers.
XT=3	Sinusoidal brushless control with current commutation. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board.
XT=5	Three phase AC induction motor control. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board. A standard 300V (and above) driver for a three phase brushless motor that accepts two phase command is also suitable for induction motor.

3.2 SB214PC2 (dual axis)

3.2.1 Initial setting

The SB214PC2 controller has factory initiated setting that may be changed according to your needs:

PC bus address:	100Hex.
Interrupt:	disabled.
Drive command type:	PWM (0-5V).

Isolation:	Full isolation between the drives and the controller.
Motor/Drive type:	Standard servo with single command.

If this setting does not fit your needs, see SB214PC2 PC Based Dual Axis Controller, page 5-20.

3.2.2 PC Bus Address

The SB214PC2 are factory set to communicate over the PC bus at I/O address 256 decimal (100 hex). The address is controlled by the setting of the dip switch SW8-SW2 (A9-A3). Unless there is a conflict at this address it is recommended to use the default address. To change the address see Address Setup, page 5-33.

3.2.3 Interrupt

The cards are set at the factory to communicate with the PC without causing interrupts. The ACS Adjuster ® for Windows does not use the interrupt capability and all the interrupt setup jumpers must be left opened. To choose an interrupt see Interrupt Setup, page 5-34.

3.2.4 Drive Command Type

Two command types are supported: PWM (0-5V) type and analog type (± 10V). The type is chosen by JP11 through JP14.
See Driver Interface, page 5-34

3.2.5 Isolation

The card can be completely isolated from the drives (including the analog commands) and the digital I/O. Set JP5 - JP10 to select whether you want an isolation from the Drivers, Inputs, Limits, Emergency Stop and/or Outputs. See Isolation And External Supplies, page 5-34.

3.2.6 Using The Card Outside The PC (with serial communication).

To use the SB214PC2 as a stand-alone outside the PC, set SW1 dip switch to 'off'. See Address Setup, page 5-33.

3.2.7 Installing The Board In A PC

The SB214PC2 can be installed in a 32, 16 and 8 bits PC bus.

Place the SB214PC2 in a free slot. Make sure that there is enough space for the cables that are needed for the interface. You may leave the adjacent slot free and use its rear exit for the flat ribbon cables.

Fasten the board rear plate with a screw.

3.2.8 Connecting To The SB214PC2INT Distribution/Interface Box

Install the flat ribbon cables J6 (rear), and J4.

SB214INT2	SB214PC2
JA	J6
JB	J4

3.2.9 Powering Up The System

Power up the PC and the external supplies. The on board LED should blink for a short period.

Assuming that the The ACS Adjuster ® for Windows is already installed on your PC, activate it, set the address and switch to terminal mode.
While in terminal mode, press **<cr>**. The controller should respond with the **0>** prompt. Now type **RG1<cr>**. The controller should respond with a list of all the parameters of Group #1.

3.2.10 Motor Types

The motor type is set by the XT (aXis Type) parameter. Change the type only when the axis is disabled. Afterwards execute a **SAVE** command and then a **HWRES** command.

XT=0 Stepper control with current commutation (with DR170 drivers made by ACS) - Two command signals per axis.

XT=1 Standard DC servo control with one command signal per axis.

XT=2 Stepper control. Pulse-Direction pair of signals per axis. Can be interfaced to standard stepper/micro-steppers drivers.

XT=3 Sinusoidal brushless control with current commutation. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board.

XT=5 Three phase AC induction motor control. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board. A standard 300V (and above) driver for a three phase brushless motor that accepts two phase command is also suitable for induction motor.

3.3 SB214VME

3.3.1 VME Bus Address

The VME bus has 32 address lines A31-A0. Lines A31-A16 are defined by the VME system. Those lines do not reach the controller. Within the 64K address space that is defined by the VME system, the card offset is defined by A15-A8 via the on board DIP switch (A15-SW1, A8-SW8). **A15 (SW1) must be set to "on" (when SW1 is "off" the card communicates via the serial link)**. The SB214VME are factory set to communicate over the VME address 00H. Within the 256 bytes address space that is defined by A15-A8, the addresses of the four communication registers are:

Add.	Register
61H	TX_FIFO, Read from controller's FIFO.
41H	RX_FIFO, Write to controller's FIFO.
21H	FLAGS, Read controller's flags register.
01H	STATUS, Read controller's status register.

FLAG - Flags register bits:

D15 - D4	D3	D2	D1	D0

Bit	Flag
D0	TX_FIFO Full.
D1	TX_FIFO Empty.
D2	RX_FIFO Full.
D3	RX_FIFO Empty.

The flags are active low.

3.3.2 Drive Command Type

Two command types are supported: PWM (0-5V) type and analog type (± 10V). The type is chosen by JP15 through JP18.
See Drive Interface, page 5-36.

3.3.3 Isolation.

The card should be completely isolated from the drives (including the analog commands) and the digital I/O. Set JP19, JP20, JP21 to select whether you want an isolation from the Drivers, Inputs, Limits, Emergency Stop and/or Outputs. See Isolation And External Supplies, page 5-55.

3.3.4 Installing The Board In A VME Rack

Insert the SB214VME in a free slot in your VME rack.

Fasten the board front panel with the top and bottom screws.

3.3.5 Connecting The Cables

You can use the front flat cable connector and/or the rear P2 lower connector, according to the signals you need for your system.

3.3.6 Powering Up The System

Power up the VME rack and the external supplies.

The SB214VME front LED should blink for a short period.

The system is ready for use.

3.3.7 Motor Types

The motor type is set by the XT (aXis Type) parameter. Change the type only when the axis is disabled. Afterwards execute a **SAVE** command followed by **HWRES** command.

XT=0	Stepper control with current commutation (with DR170 drivers made by ACS) - Two command signals per axis
XT=1	Standard DC servo control. One command signal per axis.
XT=2	Stepper control. Pulse-Direction pair of signals per axis. Can be interfaced to standard stepper/micro-steppers drivers.
XT=3	Sinusoidal brushless control with current commutation. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board.
XT=5	Three phase AC induction motor control. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board. A standard 300V (and above) driver for a three phase brushless motor that accepts two phase command is also suitable for induction motor.

3.4 SB214ND

3.4.1 RS-232/422 Interface

Set the host/terminal for full duplex, 8 data bits, no parity, one start and one stop bit. Connect the RS232 cable to the unit. It is recommeded to use the ACS Adjuster for Windows for proper setup.

3.4.2 Powering Up The System

Power up the unit. The on board LED should blink for a short period.

Assuming that the ACS Adjuster® for Windows is already installed on your PC, activate it, set the address and then switch to terminal mode.
While in terminal mode, press **<cr>**. The controller should respond with the **0>** prompt. Now type **RG1<cr>**. The controller should respond with a list of all the parameters of Group #1.

3.4.3 Drive Command Type

Two command types are supported: PWM (0-5V) type and analog type (±10V). See Drive Interface, page 5-73.

3.4.4 Motor Types

The motor type is set by the $XT (aXis Type) parameter. Change the type only when the axis is disabled. Afterwards execute a **SAVE** command and then a **HWRES** command.

XT=0	Stepper control with current commutation (with DR170 drivers made by ACS) - Two command signals per axis.
XT=1	Standard DC servo control with one command signal per axis.
XT=2	Stepper control. Pulse-Direction pair of signals per axis. Can be interfaced to standard stepper/micro-steppers drivers.
XT=3	Sinusoidal brushless control with current commutation. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board.
XT=5	Three phase AC induction motor control. Two phase current commands per axis. Requires a driver that accepts two phase current commands and generates the third (and fourth) command on board. A standard 300V (and above) driver for a three phase brushless motor that accepts two phase command is also suitable for induction motor.

3.5 Working with The Controller

Direct Mode

In this mode the host terminal communicates with the controller via the communication link. Each command is interpreted and executed immediately as it is typed.

Warning - When you activate the motor for the first time, disconnect it from external loads and make sure that the area is clear of any object that might be hit by the moving motor !!!

Making A Point To Point Move

SXMO1MM0<cr>	Enable the motor, and choose mode 0 (ptp)
SXRP50000<cr>	Set a Relative Position move of 50000 counts
SXLV10000 LA500000 **LD50000<cr>**	Specify the Linear Velocity, Linear Acceleration and Linear Deceleration
SIC3<cr>	Set the Initiate Communication to 3 When IC=3, the controller prompts back a message whenever a move is requested, and when a move is terminated
BX<cr>	Send a Begin command The controller should prompt back with the following message: **0BX01** 0 - The ID number BX - A response to a BX command 01 - The result code. 01 means a successful operation If the move is executed successfully, the controller prompts with the following message: **0EX01** If the End message is different than 01, check the meaning of the error code in the **ACS Software Guide**, The last of Begin & End messages can also be retrieved using the T1, T2 commands. Try it: **T1 T2<cr>.**

Summary

To make a move you should activate the motor(s), set the mode, the required velocity and acceleration, the target, and then issue a Begin command. If something goes wrong, the error messages should help you to find the reason of failure. To retrieve the value of any parameter, send a report command, which consists of the letter R followed by the axis notation (X), then the mnemonics of the required parameter(s).

Making A Point To Point XY Interpolated Move

SXMO1 MM0<cr> **SYMO1 MM0<cr>**	Enable the motors, and choose mode 0 (ptp)
SXRP50000 <cr> **SYRP35000<cr>**	Set a Relative Position move of 50000, 35000 counts
SVV10000<cr> **LA500000<cr>** **SVD500000<cr>**	Specify the Vector Velocity, Acceleration and Deceleration
BXY<cr>	Send a Begin command

3.5.2 Running Z Axis at a Constant Velocity

SZMM10 LV25000<cr>	Switch to Motion Mode 10, and set the required velocity to 25000 counts/sec You may set more than one parameter in the same command line, as demonstrated above
BZ<cr>	Start to move The motor accelerates to the desired speed.
RZAV<cr>	Find out what is the actual speed by sending a Report Actual Velocity command - The controller measures the actual distance passed during 0.01 second, multiplies it by 100, and prompts with that value.
SZLV-20000<cr>	Switch to a speed of 20000 counts/sec, in the opposite direction The motor decelerates down to the required velocity.
T0<cr>	To find out the status of the motor(s), use Tell 0 command
EZ<cr>	To stop the motor, send an End command

3.5.3 Using A Joystick

It is assumed that a joystick is attached to the X, Y analog inputs.

SXMM21<cr>	Change to manual joystick control (MM=21).
BX<cr>	Send a Begin command.
SXLV50000<cr>	Now move the joystick. The controller generates velocity commands which are directly related to the analog voltage output of the joystick potentiometer. The maximum velocity is the value of **LV**.
SXLV10000<cr>	If the motor runs too fast, and you want to have a better position control, with higher resolution, decrease the value of LV to a lower value, say 10000 counts/sec.
EX<cr>	Terminate the move by an End command.

As you can see, it is a "smart joystick". Its speed command can be adapted to the needs of the application. Once you master the **ACSPL** language you should be able to write a simple program that will allow you to switch from high speed (for long travel) to a low speed, having better position resolution and control, with the press of a switch that is attached to one of the inputs.

In order to prevent axis movement around the stationary point of the joystick, a dead band can be defined via the Lower Threshold (**LT**) and Upper Threshold (**UT**) parameters. See the Reference section in the **ACS Software Guide**.

3.5.4 Working With ACSPL Programs

ACSPL is a simple, yet powerful language that allows you to implement almost any sequence of moves, time, external input and position related events as your application may require.

Point to Point Move

In the following example the motor should make a point to point move and display simple diagnostic messages on the screen.

P<cr>	Switch to Programming mode
I<cr>	Switch to Insert mode
Enter the following statements, each one followed by **<cr>** .After typing each line, the controller prompts with the new line number	
Example:	a label
let XMO=1	Enable the motor
let XMM=0	Point to point mode
let XRP=50000	Relative distance of 50000 counts
exec BX	A Begin motion command
if X_MOVE do	Check if moving
disp " moving OK.."	
end	end of the DO block
else do	
disp "Something is wrong."	
disp " press T1T2<cr> to find the reason"	
end	
till ^X_MOVE	Wait till done
if X_END=1 do	Check for reason of termination
disp "Motion completed successfully"	
disp "New location = ", XCP	
end	
stop	End of program
After the "stop" statement press **<cr>** again. That should terminate the Insert mode. The controller prompts with a **P>**	
1,18L<cr>	Get a list of the program
C<cr>	Compile the program
X Example<cr>	Execute the program
SAVE<cr>	Save this program in the non volatile memory
SAVE<cr>	Repeat the SAVE command

This program has demonstrated how easy it is to implement an application with built-in diagnostics. The **ACS Software Guide** includes a full description of the language.

Searching For Index

One of the advanced features of the controller is its ability to search for the index of the encoder at any speed. In the following example the X motor is running at constant speed. Each time the index is passed, the position of the index is displayed. If you use a 2000 line encoder (8000 counts/rev), then you should get an index pulse exactly every 8000 counts. If you get a different number, it is a strong indication that something is wrong with the encoder feedback path (encoder, connector, encoder interface).

let XMM=10	Constant velocity mode
let XMO=1	Enable the motor
let XLV=50000	Set the velocity
exec BX	Start moving
V0=XIX	Dummy reading of the index to clear the buffer
Loop:	Label
till X_INDEX	Wait till the index flag is true
disp XIX	display the index position
goto loop	
stop	

Working With I/O

The following program reads the status of inputs 1 to 4 and sets outputs 1 to 4 accordingly.

I_O:	
V0=IP&15	Read the input port and ignore the values of inputs 5 to 16.
let OP=V0	Set the outputs.
disp "IP+",IP	Display the values of the I/O ports
disp "OP+",OP	
stop	

4. CONTROL SETUP

This section describes how to set up the control algorithm in order to operate properly AC induction, DC brushless, DC brush and step motors.

These motors and the corresponding drivers can be divided into the following groups:

Group 1	No commutation is required. This group includes all DC brush motors and drivers, AC servo motors (DC brushless and induction) where the (external) driver does the commutation, and any other actuator that generates torque or velocity which is directly proportional to a single input command. Whenever such a system is used, within the Adjuster the motor should be chosen as a "DC Brush" type ($XT=1).
Group 2	Commutation is required. This group includes three phase DC brushless and induction motors, and two and four phase step motors. The controller generates the phase current commands ("commutation"). The motor driver must be a current/torque amplifier that accepts the current commands for two phases and generates internally (by simple addition) the commands for the other phases. Whenever such a servo system is used, when working with the ACS Adjuster the motor should be chosen accordingly. E.g., "Rotary induction motor" or "Linear brushless motor". The DR170 drive box is an example of a two and four phase stepper driver that fits to this definition. The controller generates sine wave current commands with 10 bits resolution per 360 electrical degrees (=256 microsteps per full step).
Group 3	Pulse-Dir follower. This group includes all step motors with driver (translator) that accept a two signal - Pulse+Driection - command. There are also DC brushless and AC induction drivers that close the servo loop, including the position loop internally, and accept a pulse-dir type of command. The DCL10X1 universal digital driver can be operated in such a way. These systems should be referred to as "step motors with pulse-dir" ($XT=2).

Use the **ACS Adjuster** program to setup the unit to work with a servo motor. The program will guide you through all the steps necessary to set the system. Afterwards it enables you to use the stored setup and to transfer it to another controller.

4.1 About D And K Arrays

Some of the control parameters have special assigned mnemonics. **GA,** and **KZ** are two examples. Some control parameters are part of the **D** array. **D8** is the velocity loop gain, D22 is the velocity feedback gain.

To view the value of a **D** array entry use the **ADXRE** command. To set, use the **ADXSE** command. .

To report the value of **D4 - ADXRE4<cr>.**
To set **D8** to 50 - **ADXSE8 50<cr>.**

The **K** array is used for some limitations setup. For example, **K2** limits the velocity loop error to prevent overflow.

To view the value of a **K** array entry use the **AKXRE** command. To set, use the **AKXSE** command.

To report the value of **K2 - AKXRE2<cr>.**
To set **K3** to 100 - **AKXSE3 100<cr>.**

To be able to change values of **D** and **K** array, we must first remove the protection mechanism. First make a note of the present (default) value of **QP - RXQP<cr>.** Afterwards, remove the protection by setting **QP** to zero - **SXQP0<cr>.** (Trying to set a **D** or **K** parameter with the protection on will result in communication error 44). When the setup is completed and you are ready to execute a save, change **QP** to its original value **(SXQP860<cr>).** This will prevent you, or your customer from erroneously changing some sensitive control parameters.

4.2 Working With Servo Motor

4.2.1 Control Loops Block Diagram

Figure 4-1 describes the control block diagram. It consists of these components:

- Current amplifier. (Detailed in Figure 4-2)
- Velocity loop
- Position loop
- Commutation
- Coordinate transformation (F1, F2)

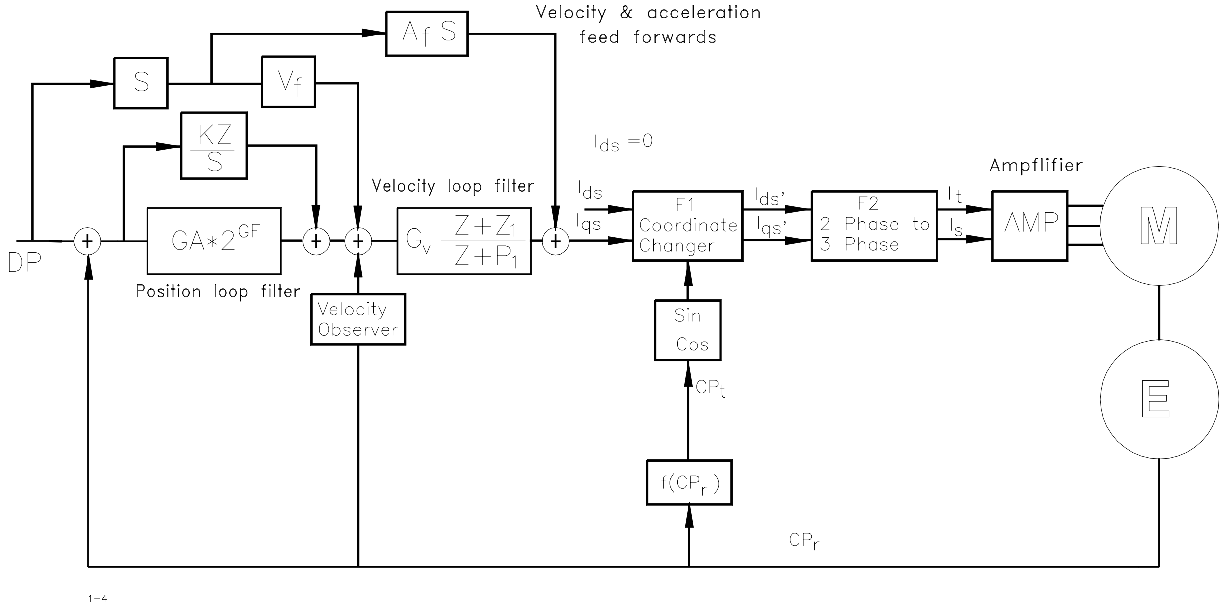

Figure 4-1 Servo motor - Control loops block diagram

If the motor-driver belong to group 2 (commutation), then the driver / amplifier must be a current (=torque) amplifier that accepts two signals command, like the one described in Figure 4-2.

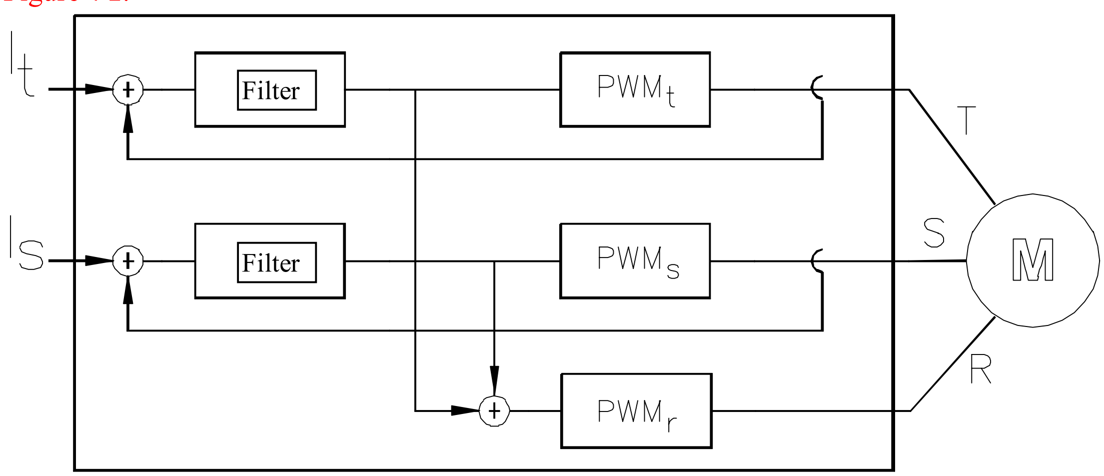

Figure 4-2 Three phase motor - Amplifier block diagram

4.2.2 Setup Procedure

Adjusting the control for a given application consists of the following steps:

Protection Setup

The protection setup goal is to protect the system and the application against misuse, over current, over heat and user's mistakes. The affects the following parameters:
Current Level ($CL)
Error limit ($ER)
Error during acceleration ($EA)

Torque Limit while not moving ($TL)
Torque limit while moving ($TO)
See the ACS Software Guide for full details. A full explanation can be found in the ACS Adjuster help on the ‘Protection Setup’ subject.

Setting Motor And Feedback Information

The following parameters are set according to the type of motor, its number of poles, and the encoder resolution:
aXis Type ($XT)
Number of Poles ($NP)
Commutation Usage ($CU). You must choose “encoder”.
Line per Rev. ($LR)
Line Factor ($LF).
See the ACS Software Guide for full details. A full explanation can be found in the ACS Adjuster help on the ‘Motor & Feedback’’ subject.

Feedback Verification

This step verifies that the safety inputs operate properly, and that the encoder is counting correctly.
When using a DC brush motor, the motor must move forward (the encoder counts up) when the Dac offset ($DA) is positive:

- Activate the terminal.
- Set the position gain ($GA) and gain factor ($GF) to 0.
- Bypass the velocity loop by setting K5 to 0 - **AK$SE5 0<cr>.**
- Enable the motor - **S$MO1<cr>.**
- Apply a positive Dac offset **- S$DA50<cr>**
  Now the motor should move in the “positive” direction, counting up. If it moves in the “negative” direction, you must either swap between the two motor wires, or between a pair of one of the encoder channels (A <-> A’).
  This will guarantee that afterwards, when the velocity and position loops are closed, that the feedback polarity is the correct one -negative.
  During this process the motor might trap on error, generating end error message 25.

Commutation Setup

This step is not needed when using a group 1 type system (DC brush). The goal of the commutation adjustment is to align the magnetic field with the position feedback reading. It is done automatically by the ACS Adjuster. When using a brushless motor, the procedure must be executed after each power up. The ACSPL **214COMST.PRG** program (which you should get from the distributor) does it.

- Load the program.
- Compile it **(C<cr>).**
- Save the present setup into the non-volatile memory **(SAVE<cr>)**

- Execute h/w reset (**HWRES<cr>)**.

Now after each power up (as part of the AUTOEXEC) execute it.

When using an AC induction motor, the goal of the commutation adjustment is to align the motor windings orientation with the reading of the feedback device and to properly set the **F**ield **C**urrent (**FC**) and **S**lip **C**onstant (**SK**) values. This procedure must be executed one time only.

Velocity Loop Setup

The velocity loop includes two filters: a forward filter, and a velocity observer. In most cases it is sufficient to adjust only the forward loop filter. In some systems which have a low frequency mechanical resonance (belt driven systems, direct drive systems with high inertia), you will be able to overcome the negative effect of the resonance by adjusting the bandwidth of the velocity observer.

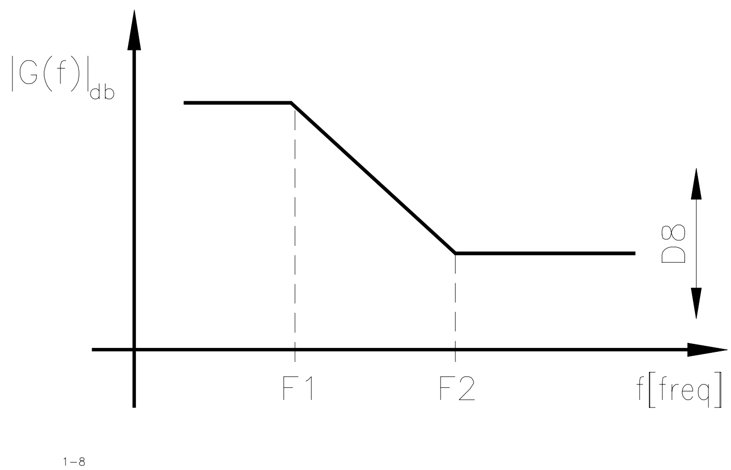

Figure 4-3 Velocity loop forward filter - Bode diagram

The forward filter is a lag type of filter. Its gain is **D8.**
The two break point frequencies (Figure 4-3) - F1 and F2 define **D5, D6, D7.**

$$D5 = \frac{1+\dfrac{2}{0.00005\cdot 6.28\cdot F2}}{1+\dfrac{2}{0.00005\cdot 6.28\cdot F1}}\cdot 2^{16}$$

$$D6 = -\frac{1-\dfrac{2}{0.00005\cdot 6.28\cdot F2}}{1+\dfrac{2}{0.00005\cdot 6.28\cdot F2}}\cdot D5$$

$$D7 = -\frac{1-\dfrac{2}{0.00005\cdot 6.28\cdot F1}}{1+\dfrac{2}{0.00005\cdot 6.28\cdot F1}}\cdot 2^{16}$$

D22 is the scale factor between the maximum velocity and its internal representation. In order to get the best dynamic range for the velocity measurement, you must set D22 accordingly:

$$D22 = \frac{1638 \cdot 60 \cdot 20000}{Max\_Vel \cdot Ke} = \frac{1{,}965{,}600{,}000}{Max\_Vel \cdot Ke}$$

Where:

Ke - Encoder's resolution. Counts/rev. For example - A motor is equipped with 1000 line encoder and there is no external multiplier. Ke=1000*4=4000 (4 is the internal multiplier).

Max_Vel - 20% above the maximum velocity used in the application, or 60% of the maximum specified velocity of the motor, which ever is higher.

Units - rpm.

The motor's maximum velocity is 6000rpm. The maximum velocity used is 4000 rpm. Therefore, Max_Vel=4800.
Assuming that Ke=4000, D22=102
(AD XSE22102<cr>).

The motor's maximum speed is 2000rpm. The maximum velocity used is 200rpm. Max_Vel=1200 rpm.
Assuming that Ke=4096, D22=400
(AD XSE22 400<cr>).

The velocity feedback signal is derived from the encoder reading by an observer.

The velocity observer bandwidth is factory set to 1000Hz.
Sometimes, 1000Hz might be too high. You may hear high frequency audible noise. If necessary, the bandwidth can be reduced by changing **D9, D10** according to Table 4-1.

F0 - Observer bandwidth [Hz]	SA[sec]	D9	D10
1000	0.00005	29117	6468
500	0.00005	14558	1617
250	0.00005	7279	404
200	0.00005	5823	259
150	0.00005	4368	146
100	0.00005	2912	65
70	0.00005	2038	32
50	0.00005	1456	16

Table 4-1

Sometimes, the system may include a low frequency resonance mode. In this case you may find it difficult to achieve a stiff and stable system. Again, reducing the bandwidth of the velocity observer may solve this problem by attenuating the resonance frequency.

$$D9 = \sqrt{2} \cdot 0.00005 \cdot 6.28 \cdot F0 \cdot 2^{16}$$

$$D10 = (0.00005 \cdot 6.28 \cdot F0)^2 \cdot 2^{16}$$

Where F0 - The observer bandwidth

Reducing the bandwidth too much might impair the stability of the velocity loop.

Position Loop Setup

The position loop filter is a PI type filter. The proportional gain is GA•2^{GF}. The Integrator gain is KZ. See Figure 4-4.

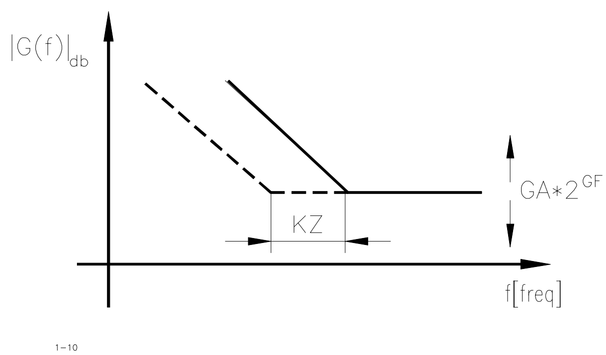

Figure 4-4 Position loop filter

Every 50 microseconds the following calculation is made:

Y1(n) = Y1(n-1) + (Kz/100)*Pe ;Integrator

Y2(n) = GA•2^{-GF}•Pe ;Proportional gain

Y(n) = Y1(n)+Y2(n)

The adjustment is executed while monitoring the actual velocity profile and position error response to point to point trapezoidal move.
The goal is to achieve a smooth velocity response without overshoots and high frequency noise while keeping the following error to a minimum value.

Acceleration Feed Forward Setup

The Acceleration Feed forward (AF) setup reduces the following error during accel/decel periods.
If AF is too large, the error may become excessive and the motor will become noisy.

Optimizing Torque At High Speed - Phase Advance

(For DC brushless motors, group 2 only). The phase advance feature improves the torque-velocity characteristics at high speed. At high speed, the actual current lags behind the command. As a result, the motor either needs more current to produce the required torque, or cannot produce the required torque at all. (Figure 4-5)

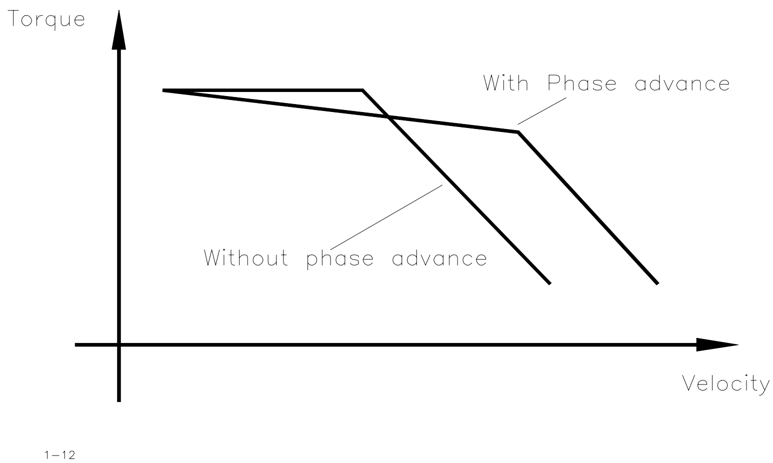

Figure 4-5 DC brushless motor - Velocity vs. torque

4.3 Working With Step Motor

4.3.1 Pulse-Dir Drives

The controller can generate Pulse & Direction commands to feed a standard stepper or microstepper driver. The maximum pulse frequency is 2MHz.

- Set the aXis type to 2 (Pulse-Dir).
- Set the Pulse Width according to the drive specification.
  The Pulse Width ($PW) is programmable in the range of 400 nanoseconds, up to 60 microseconds.
  If $PW[sec]•$LV[pulse/sec] ≥ 1, a begin error 24 is generated when a begin command is issued.
- Execute a SAVE and HWRES.

4.3.2 ACS Drives (S/W commutation)

When using a driver like the ACS DR170, the controller generates the sine and cosine current commands to the driver. The command resolution is 256 microsteps per full step. The drive itself consists of two H bridge pwm amplifiers.

You must set the aXis Type ($XT) to 0 **-S$XT0<cr>**

Current Level Setup

The phase current level is programmable. It can be individually programmed for different stages of the move profile. There are six parameters that control the various states of current level:

(I_c - Nominal phase current of the drive - 6.5A)

CL Current Level. 0 - 100%.
The maximum current is I_c •CL/100

P0 Idle current level. 0 - 150%.
Idle current = I_c •CL•P0/100². The maximum is limited to I_c.

P1 Jerk duration current level. 0 - 150%.
Jerk current = I_c •CL•P1/100². The maximum is limited to I_c.

P2 Acceleration current. level. 0 - 150%
Acceleration current = I_c •CL•P2/100². The maximum is limited to I_c.

P3 Slew speed current level. 0 - 150%.
Slew speed current = I_c •CL•P3/100². The maximum is limited to I_c.

P4 Deceleration current level. 0 - 150%.
Deceleration current = I_c •CL•P4/100². The maximum is limited to I_c.

The motor is specified for 3.0A per phase. We want to set the idle current to 1.5A, and the move current to 3A:
CL = 100•3/6.5=46%
P0 = 100•1.5/3=50%
P1,P2,P3,P4=100
SXCL46 P050 P1100 P2100 P3100 P4100<cr>

After you finish to set the parameters, SAVE them into the non volatile memory - **SAVE<cr>** twice.

Using The Stepper With Encoder Feedback

The encoder can be used for both stall detection position verification and correction.

The ratio between the number of encoder counts/rev and the number of microsteps/rev is programmable by the use of two parameters:

Gear ratio = $XEC/2^{XEF}$

XEC - verification Encoder Constant [1 - 32,767]

XEF - verification Encoder Factor [0 - 23]

Using a stepper with 50,000 microsteps/rev, and a 1000 line quadrature encoder (= 4000 encoder counts/rev), the gear ratio is 12.5:
XEC=25, XEF=1

Verify Enable (XVE) parameter.

The verification encoder related features are activated by the use of **XVE** parameter.

XVE=0	No encoder feedback is used. The Current Position (XCP) is derived from the generated command, and the Position Error (XPE) is always zero. The velocity monitoring signal derived from the generated profile.
XVE=1	The verification function is activated. The Current Position (XCP) reflects the position of the encoder in microsteps units as defined by the gear ratio. The velocity monitoring signal is derived from the encoder feedback. Upon setting the XVE value to 1, XCP gets the value of the present XDP (present Desired Position): XCP <-- XDP, and the Position Error (XPE) is therefore zero (XPE = XDP-XCP). XVE can be set to 1 only when the axis is not in motion. When the verification function is active, the Position Error (XPE) is set to zero whenever the Step Resolution (XSR) is changed, and when the SXZP command is used.
XVE=2	Upon setting the XVE value to 2, XDP gets the (translated) value of the encoder position: XDP <-- XCP. Once the encoder is initialized correctly, it is tracking the actual position of the axis even if the motor lost synchronization. When the stepper loses synchronization, it is recommended to use XVE=2 in order to reset XDP to fit the location as measured by the encoder. XVE can be set to 2 only when the verification is already active and the axis is not in motion.

Stall Detection

Whenever **XVE>0**, and (**|XPE| > XER**), and error message 25 is generated, the move is terminated, and either **AUTO_ERR** or **AUTO_FLT**, user defined programs, is activated

X ERror limit = 5000 (one tenth of a rev.). If the **P**osition **E**rror (**XPE**) has exceeded 5000, the following program will be automatically activated:
AUTO_ERR:
disp "Motor lost position"
ret

Position Correction

When **XVE>0**,and a motion is executed, once the profile generation is done, the controller starts an internal clock. After **XTD** milliseconds, the controller checks if the Position Error (**XPE**) is smaller then the Target Radius (**XTR**). If not, the **AUTO_EXX** routine is activated.

XTR=125, XTD=25, and the motor was commanded to move to the origin by setting **XAP=0**. Assuming that at the end of the move **XCP=300**. The following routine will automatically be executed, trying to correct the position:

V0=XPE	
V0=XPE	Record the error
let XVE=2	XDP should reflect the position of the encoder.
let XRP=V0	Move the difference
exec BX	
till ^X_MOVE	
ret	

Note that if the correction move happens to be insufficient, **AUTO_EXX** is called again to correct the correction.

5. ELECTRICAL INTERFACE

5.1 SB214PC 4 Axis PC Based Controller

The **SB214PC** can be interfaced directly to external drives, encoders, I/O, etc., using flat ribbon cables. All signals are buffered and/or isolated on board. The SB214INT is an optional distribution box. Its goal is to simplify the interfacing by using Phoenix type connectors with a dedicated connector for each function. The controller interfaces with the external world via 4 flat ribbon cables, connectors: J1, J2, J4, J5.

The description of the SB214INT interface is identical to the description of the SB214ND interface. See SB214INT Distribution Box, page 5-18.

5.1.1 Power Supply

All drive interfaces and I/Os signals are opto-isolated. To maintain the isolation, an external supply must be used to supprt the isolated, user's side. The non isolated side is supplied by the PC supply via the PC bus.

Controller Non Isolated Supply

The PC Bus supply. These voltages appear in the Flat Cable Connector:

COMMON

+5V

+12V

-12V

+5VFU (+5V through 1.6A fuse)

User Isolated Supply

An external supply:

RTNF (Field common)

+5VF

+12VF

-12VF

These voltages can be supplied from the PC (thus loosing the isolation) by using the jumpers J25, J27, J28 and J29. See page 5-15.

5.1.2 Drive Interface

The controller can control the following types of motors/drivers:

- A stepper motor driver with two phase current commands (XT=0).
- A servo motor with single torque/velocity command (XT=1).
- A stepper motor driver with Pulse-Dir command (XT=2).
- A sinusoidal brushless motor with a drive that accepts two phase current commands (the controller does the commutation) (XT=3).
- An AC induction motor with a drive that accepts two phase current commands, using Field Oriented Control (Vector Control) method (XT=5).

The interface consists of the following signals:

PWM0,PWM1 (or V0,V1), Enable, Fault, Settling, RTNF (return).

PWM0 (V0), PWM1 (V1) Servo Drive Commands

The command is available in two forms:

PWM	0-5V level. 50% duty cycle represents zero command.
Analog	±10V. It is generated by passing the PWM signal through a low pass filter.

The desired signal is selected by jumpers JP17 to JP24.

See page 5-15.

Offset Adjustment

The offset of V0, V1 can be adjusted using the following potentiometers:

Axis	Signal	Potentiometer
X	V0	RV1
X	V1	RV2
Y	V0	RV3
Y	V1	RV4
Z	V0	RV5
Z	V1	RV6
T	V0	RV7
T	V1	RV8

Pulse-Dir Drive Command

When using a driver that accepts Pulse-Dir input (step or servo motor), then the PWM0,1 designated commands must be used, PWM0 is the Pulse (clock) signal, and PWM1 is the Dir signal.
The pulse width is programmable by the Pulse Width ($PW) parameter in the range [0.4 - 60] microsecond.The pulse rate range is [1 - 2,000,000] pulses/sec.

The aXis Type ($XT) must be set to 2. Then the controller outputs the Pulse-Dir signals via the PWM0,1 designated outputs.

JP17-JP24 select the type of command for a specific axis.

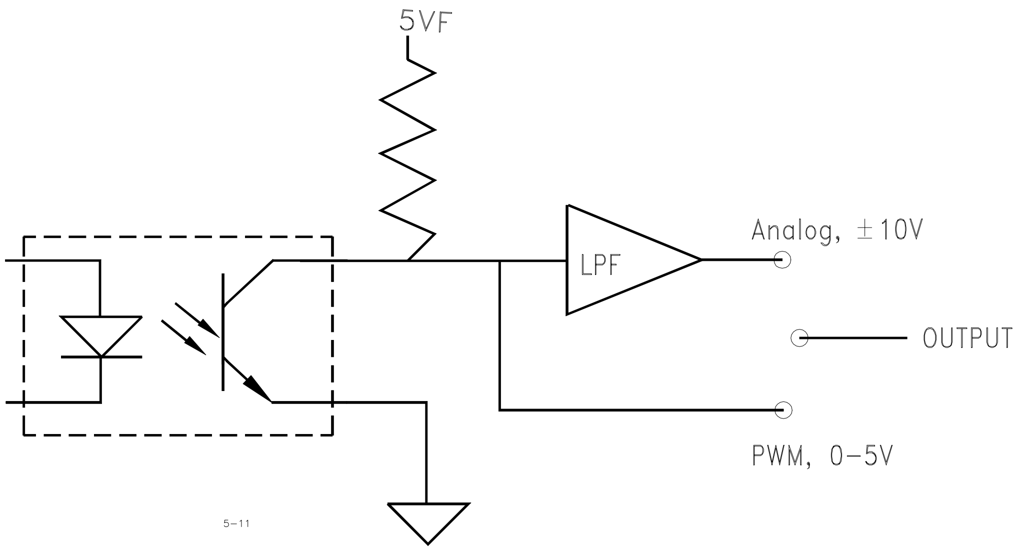

Figure 5-1 PWM / analog command selection

Enable (Output)

The enable signal polarity is programmable via the $OL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that are specified by the Enable Delay $ED) parameter before it checks the status of the drive fault feedback signal.

The Enable output is isolated. See Figure 5-2.

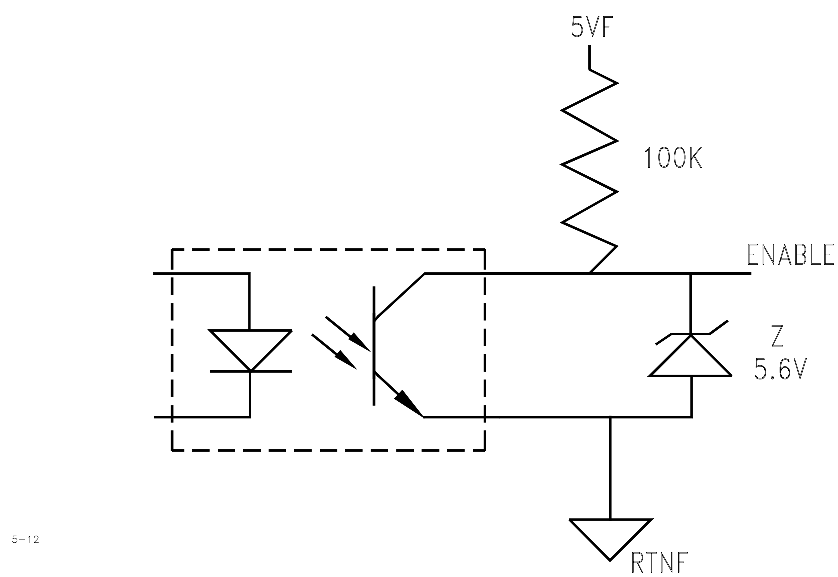

Figure 5-2 SB214PC Drive enable

Fault (Input)

The Fualt signal polarity is programmable via the $IL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that are specified by the Enable Delay ($ED) parameter before it checks the status of the drive fault feedback signal.

When the fault feedback becomes TRUE, an error message E26 is generated.

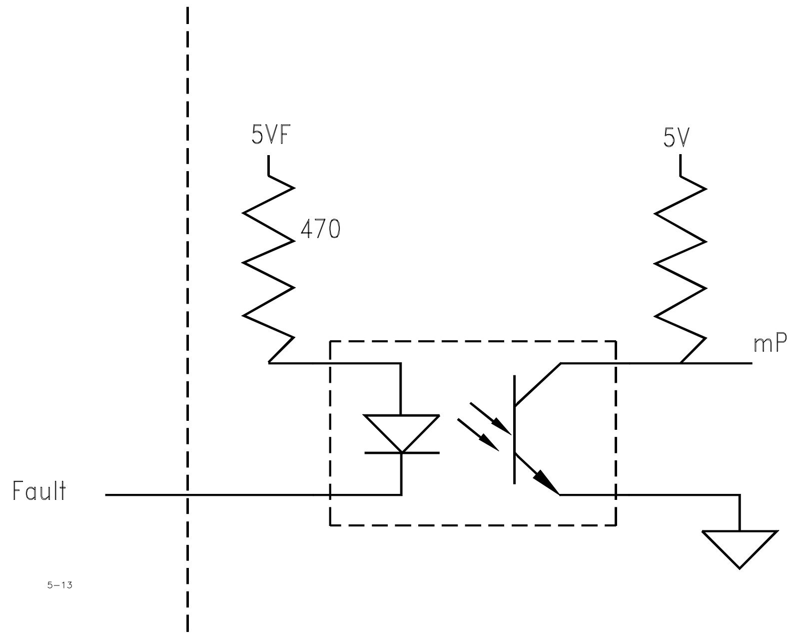

Figure 5-3 SB214PC Drive Fault

Settling

This signal is used with the DR170 micro-stepper drivers for four phase stepper motors. When enabled, it shorts two out of the four phases, thus producing a significant damping effect.

The electric circuit of the Settling Output is identical to that of the Enable output.

5.1.3 Encoders

There are six encoder ports - encoder #1 (primary) for X,Y,Z and T, and encoder #2 (secondary) for Z and T. Each can serve a different function according to the setup of the specific axis.

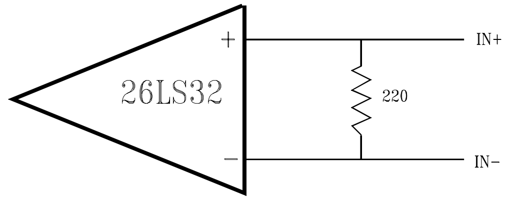

Figure 5-4 Encoder interface

Each encoder input port accepts three channel differential, TTL level, optical encoder, or laser interferometer signals of various formats. The format and the maximum counting speed is programmable via the Encoder Type (**ET** - primary, **et** - secondary) parameters.

- UP-DOWN plus index.
  The maximum counting rate is either 10 MHz ($ET=0), or 20 MHz ($ET=100).
- CLOCK-DIR plus index.
  The maximum counting rate is either 10 MHz ($ET=1), or 20 MHz ($ET=101).
- Two adjacent bits (C0, C1) of an up-down counter plus index.
  The maximum counting rate is either 10 MHz ($ET=2), or 20 MHz ($ET=102).
- Two phase quadrature plus index. (Factory default, 10MHz).
  The maximum counting rate (including X4) is either 10 MHz ($ET=3), or 20 MHz ($ET=103).

The input buffer is built around AM26LS32 line receivers. It is recommended to use encoders with built-in line drivers (AM26LS31 or similar).

The encoder can be powered by the PC 5V via the +5VFU (fused) pin, or externally by an external supplied source. When an external supply is used, its return must be connected to the (PC) non-isolated common.

CHA+	Encoder A non-inverted input (CLOCK,UP,C0)
CHA-	Encoder A inverted input (CLOCK,UP,C0)
CHB+	Encoder B non-inverted input (DIR,DOWN,C1)
CHB-	Encoder B inverted input (DIR,DOWN,C1)
CHI+	Encoder I non-inverted input
CHI-	Encoder I inverted input

5.1.4 Input Port

There are 16 general purpose inputs. Some inputs may serve dedicated functions. See **Input & Output Ports**, page 7-1.
All are opto-isolated. (Figure 5-5). The registration mark inputs (6,7,9,10) use fast, 10MHz opto-couplers.

The fast registration inputs' filter (RC) has a time constant of 1 µsec. The rest of the inputs have a time constant of approximately 1 msec.

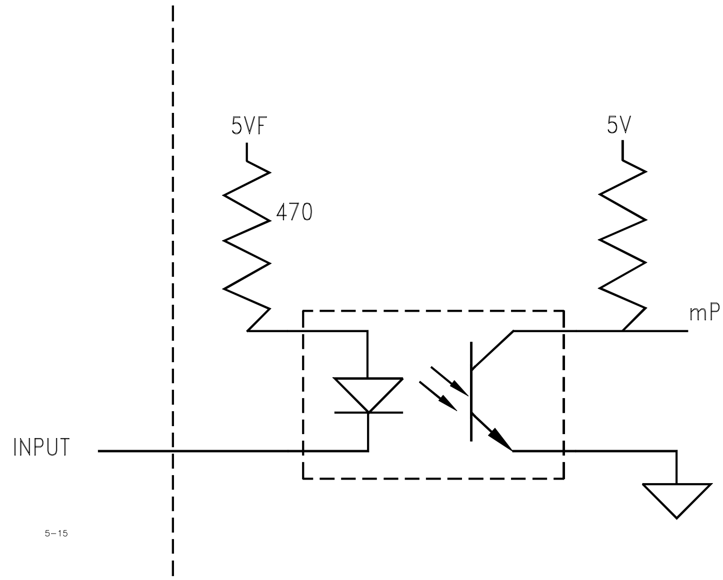

Figure 5-5 SB214PC Input port

5.1.5 Safety Inputs

The EMERGENCY STOP input is used to stop and deactivate the axis instantaneously.

The limit inputs are used to avoid axis over travel by preventing further motion in the inhibited direction.

The structure of the safety inputs is identical to the structure of the general purpose inputs.

The polarity of the safety inputs can be altered by the Input Logic (IL) parameter.

5.1.6 Output Port

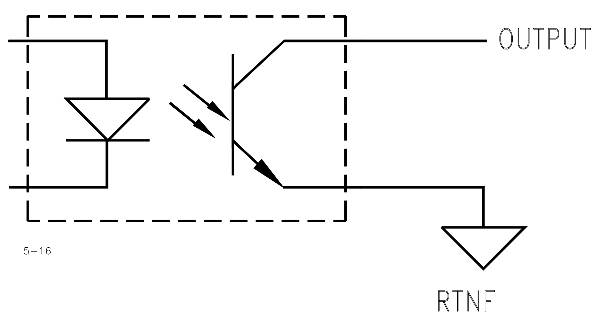

Figure 5-6 SB214PC Output port

There are 16 general purpose outputs. Outputs can be programmed to serve a dedicated function. See **Input & Output Ports**, page 7-1.

All outputs are opto-isolated with open collector. See Figure 5-6.

Outputs 1,2,3,8,9,10,11,16 use fast 10MHz opto-couplers. Each of these outputs can sink 10mA. The output level must be limited to 0 - 5V range.

Outputs 4,5,6,7,12,13,14,15 use slow opto-couplers (10-50kHz). Each of these outputs can sink at least 2.5mA. The output level must be limited to 0-24V range.

5.1.7 High Speed Serial Interface ("Interferometer")

There are two, general purpose high speed synchronous serial interface ports. They can be used for high speed laser interferometers position reading, or any other function as required by the user. Consult factory if you need to use them. They are labeled X and Y ports #2. The interface is implemented using 3 differential signals: Start, Clock and Data In. See Figure 5-7.

Name	Description
COM	Common Ground
START+	Start (Output)
START-	Inverted Start (Output)
CLOCK+	Clock (Output)
CLOCK-	Inverted Clock (Output)
DATA IN+	Data In (Input)

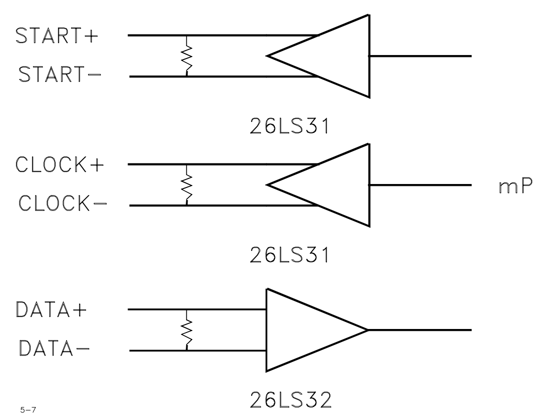

Figure 5-7 SB214PC high speed interface

5.1.8 Analog / Joystick Inputs

There are 6 Analog Inputs - AI1 - AI6. Analog Inputs 1 (ACSPL's A0) and 2 (ACSPL's A1) can be used as joystick inputs. AI1 for X and Z, and AI2 for Y and T.

The joystick potentiometer must be connected between the 5R pin and the COM pin. The potentiometer output is connected to AI1 and AI2 pins.
The two inputs act also as general purpose analog inputs that are sampled via a 10 bits ADC by the A0 and A1 ACSPL variables. 0V is represented by 0,

and 5V is represented by 1024. The user must make sure that the input voltage does not exceed the specified range of [0 - 5V].

AI3 -AI6 are ±10 V inputs that are sampled by a 12 bit ADC. 10V is represented by 2.048. -10V is represented by -2.0448.

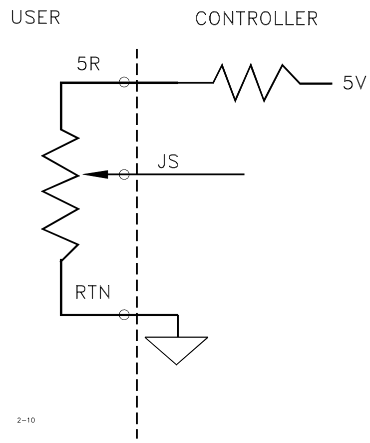

Figure 5-8 Joystick connection

Name	Description
5R	5V Supply to Joystick via 100 Ohms.
AI6	Analog Input 6. +/-10V, 12 bits [-2,048..2,047]
AI5	Analog Input 5. +/-10V, 12 bits [-2,048..2,047]
AI4	Analog Input 4. +/-10V, 12 bits [-2,048..2,047]
AI3	Analog Input 3. +/-10V, 12 bits [-2,048..2,047]
AI2	Analog Input 2. +/-10V, 10 bits [-512 .. 511]
AI1	Analog Input 1. +/-10V, 10 bits [-512 .. 511]
COM	Common Ground

5.1.9 Analog Outputs

There are five Analog Outputs, in the range ±10 V.
The analog output (**A_OUT**) is an 8 bit general purpose that can be set via the **D0** parameter.
There is one 10 bits analog output for each axis, to be used for monitoring purposes, or as an uncommitted, general purpose analog output - **X_A (XD4), Y_A(YD4), Z_A(ZD4), T_A(TD4).**

To use an axis output as an uncommitted output, set $MN to 0. For example, to use XD4 - **SXMN0<cr>.**

To use an axis output to monitor a real time signal, set $MN to 1 (velocity), 2 (current command), or 3. When $MN=3, $DC points to the required parameter. For example, to monitor the position error of Y axis - **SYMN3 DC4<cr>.** See **Motion Monitoring**, page 8-1.

Name	Description
COM	Common (ground)
T_A	T Analog Out - TD4
Z_A	Z Analog Out - ZD4
Y_A	Y Analog Out - YD4
X_A	X Analog Out - XD4
A_O	General purpose Analog Output - D0

5.1.10 Serial Communication

Asynchronous serial communication to host.

To communicate exclusively through the serial link, SW1 dip switch must be in 'OFF' position.

While communicating through the PC bus, the controller also listens to the serial link. When a SFM<cr> command is received, the controller switches to communication through the serial link only.

RXD	RS232 receive.
TXD	RS232 transmit.
M_SD	Communication shutdown. TTL level.

5.1.11 Extension Serial Interface

Interface to extension modules. (Consult factory)

S_RXDI	Receive. TTL level.
S_TXDO	Transmit. TTL level.
S_SD	Shut down. TTL level
IO_S0 - IO_S4	Select lines. TTL level.

5.1.12 Isolation And External Supplies

The drivers, the input port and the output port can be isolated from the controller/PC by setting the appropriate jumpers and using external isolated supplies. The following matrix describes the relation between the setting of the jumpers and the various isolations.

Drivers	Input port	Output port	JP25	JP27	JP28	JP29	JP14	JP15	JP13
IS	IS	IS	OFF	OFF	OFF	OFF	2-3	2-3	2-3
IS	IS	NI	OFF	OFF	OFF	OFF	1-2	1-2	2-3
IS	NI	IS	OFF	OFF	OFF	OFF	2-3	2-3	1-2
IS	NI	NI	OFF	OFF	OFF	OFF	1-2	1-2	1-2
NI	NI	NI	ON	ON	ON	ON	1-2	1-2	1-2

IS - Isolated

NI - No Isolation

5.1.13 J1 Connector

20 pin flat ribbon cable.

Pin	Name	Description
1	+12V	PC +12V
2	-12V	PC -12V
3	+5V	PC 5V
4	Common	PC common
5	RXD	RS232 RXD
6	TXD	RS232 TXD
7	M_SD	RS232 Shut Down command (TTL)
8	S_SD	Extension communication shut down (TTL)
9	S_RXDI	Extension communication RXD (TTL)
10	S_TXDO	Extension communication TXD (TTL)
11	AI3	Analog input 3
12	AI4	Analog input 4
13	AI5	Analog input 5
14	AI6	Analog input 6
15	IO_S0	Extension interface select line
16	IO_S1	Extension interface select line
17	IO_S2	Extension interface select line
18	IO_S3	Extension interface select line
19	IO_S4	Extension interface select line
20	M_LED	µP ON LED command (TTL)

5.1.14 J2 Connector

34 pins flat ribbon cable.

Pin	Name	Description
1	+5VFU	5V Fused
2	GND	Ground
3	X_DI+	X interferometer data input+
4	X_DI-	X interferometer data input-

Pin	Name	Description
5	X_CONVST+	X interferometer start+
6	X_CONVST-	X interferometer start-
7	X_SCLK+	X interferometer clock+
8	X_SCLK-	X interferometer clock-
9	Y_DI+	Y interferometer data input+
10	Y_DI-	Y interferometer data input-
11	Y_CONVST+	Y interferometer start+
12	Y_CONVST-	Y interferometer start-
13	Y_SCLK+	Y interferometer clock+
14	Y_SCLK-	Y interferometer clock-
15	Z_CH2I+	Z encoder #2 index+
16	Z_CH2I-	Z encoder #2 index-
17	Z_CH2A+	Z encoder #2 A+
18	Z_CH2A-	Z encoder #2 A-
19	Z_CH2B+	Z encoder #2 B+
20	Z_CH2B-	Z encoder #2 B-
21	T_CH2I+	T encoder #2 index+
22	T_CH2I-	T encoder #2 index-
23	T_CH2A+	T encoder #2 A+
24	T_CH2A-	T encoder #2 A-
25	T_CH2B+	T encoder #2 B+
26	T_CH2B-	T encoder #2 B-
27	X_A	X analog out
28	Y_A	Y analog out
29	Z_A	Z analog out
30	T_A	T analog out
31	AI1	Analog input 1 (Joystick)
32	AI2	Analog input 2 (Joystick)
33	+5VFU	5V fused
34	Common	5V return, common

5.1.15 J3 Supply Connector

Type: Molex 5566-4. Mating: Molex 5557-4

When the controller is used outside a PC, it must be powered by an external 5V, +/-12V supply.

Pin	Name	Description
1	+12V	12V supply. 0.1A consumption.
2	-12V	-12V supply. 0.1A consumption.
3	Common	5, +/-12V common.
4	5V	5V supply. 1.5A consumption w/o external loads (encoders).

5.1.16 J4 Connector

40 pins flat ribbon cable.

Pin	Name	Description
1	5VF	Field 5V. Must be supplied externally
2	RTNF	5VF return
3	IN1	Input 1
4	IN2	Input 2
5	IN3	Input 3
6	IN4	Input 4
7	IN5	Input 5
8	IN6	Input 6
9	IN7	Input 7
10	IN8	Input 8
11	IN9	Input 9
12	IN10	Input 10
13	IN11	Input 11
14	IN12	Input 12
15	IN13	Input 13
16	IN14	Input 14
17	IN15	Input 15
18	IN16	Input 16
19	OUT1	Output 1
20	OUT2	Output 2
21	OUT3	Output 3
22	OUT4	Output 4
23	OUT5	Output 5
24	OUT6	Output 6
25	OUT7	Output 7
26	OUT8	Output 8
27	OUT9	Output 9
28	OUT10	Output 10
29	OUT11	Output 11
30	OUT12	Output 12
31	OUT13	Output 13
32	OUT14	Output 14
33	OUT15	Output 15
34	OUT16	Output 16
35	X_STL	X Settling command. ACS ST-4 stepper drivers.
36	Y_STL	Y Settling command. ACS ST-4 stepper drivers.
37	Z_STL	Z Settling command. ACS ST-4 stepper drivers.

Pin	Name	Description
38	T_STL	T Settling command. ACS ST-4 stepper drivers.
39	5VF	Field 5V. Must be supplied externally
40	RTNF	5VF return

5.1.17 J5 Connector

60 pins flat ribbon cable.

Pin	Name	Description
1	+5VFU	5V fused
2	Common	5V return, common
3	X_EN	X Enable
4	X_FLT	X Fault input
5	X_Command0	X PWM0/V0
6	X_Command1	X PWM1/V0
7	X_RL	X Right Limit
8	X_LL	X Left Limit
9	X_CH1I+	X Encoder #1 Index+
10	X_CH1I-	X Encoder #1 Index-
11	X_CH1A+	X Encoder #1 A+
12	X_CH1A-	X Encoder #1 A-
13	X_CH1B+	X Encoder #1 B+
14	X_CH1B-	X Encoder #1 B-
15	Y_EN	X Enable
16	Y_FLT	X Fault input
17	Y_Command0	Y PWM0/V0
18	Y_Command1	Y PWM1/V0
19	Y_RL	Y Right Limit
20	Y_LL	Y Left Limit
21	Y_CH1I+	Y Encoder #1 Index+
22	Y_CH1I-	Y Encoder #1 Index-
23	Y_CH1A+	Y Encoder #1 A+
24	Y_CH1A-	Y Encoder #1 A-
25	Y_CH1B+	Y Encoder #1 B+
26	Y_CH1B-	Y Encoder #1 B-
27	RTNF	5VF return
28	RTNF	5VF return
29	5VF	Field 5V. Must be supplied externally
30	5VF	Field 5V. Must be supplied externally
31	+12VF	Field 12V. Must be supplied externally
32	-12VF	Field -12V. Must be supplied externally

Pin	Name	Description
33	Z_EN	Z Enable
34	Z_FLT	Z Fault input
35	Z_Command0	Z PWM0/V0
36	Z_Command1	Z PWM1/V0
37	Z_RL	Z Right Limit
38	Z_LL	Z Left Limit
39	Z_CH1I+	Z Encoder #1 Index+
40	Z_CH1I-	Z Encoder #1 Index-
41	Z_CH1A+	Z Encoder #1 A+
42	Z_CH1AN-	Z Encoder #1 A-
43	Z_CH1B+	Z Encoder #1 B+
44	Z_CH1B-	Z Encoder #1 B-
45	T_EN	T Enable
46	T_FLT	T Fault input
47	T_Command0	T PWM0/V0
48	T_Command1	T PWM1/V0
49	T_RL	T Right Limit
50	T_LL	T Left Limit
51	T_CH1I+	T Encoder #1 Index+
52	T_CH1I-	T Encoder #1 Index-
53	T_CH1A+	T Encoder #1 A+
54	T_CH1A-	T Encoder #1 A-
55	T_CH1B+	T Encoder #1 B+
56	T_CH1B-	T Encoder #1 B-
57	ES	Emergency Stop
58	A_OUT	General purpose analog out
59	+5VFU	5V fused
60	GND	Ground

5.1.18 Jumpers SB214PC

See Figure 5-9.

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Default R/IN	Remarks
JP1	Slave µP ROM	Installed - ROM external	Removed -internal ROM	R	2-pin jumper, For factory use only
JP2	Master µP clock source	Installed - PC system clock	Removed - dedicated clock	IN	2-pin jumper, For factory use only
JP3	Master µP reset type	Master µP uses delayed reset	Master µP uses system reset	1-2	For factory use only
JP4	H/W watchdog	Disabled	Enabled	2-3	For factory use only
JP5	Master µP interrupt source	Master µP HSO.0	Slave µP HSO.1	1-2	For factory use only
JP6	Slave bus access	Via arbitration	Slaved access disabled	2-3	For factory use only (with JP7)
JP7	Master bus access	Free access	Via arbitration	1-2	For factory use only
JP8-10	Future use			R	
JP11	PPI intrpt source	Installed - Slave HSO.0	Removed - SB2500	R	For factory use only
JP12	PPI intrpt source	Installed - Master µP HSO.0	Removed - SB2500	R	For factory use only
JP13	Inputs, limits, ES pull-up supply	PC 5V (no isolation)	External 5VF (isolation)	2-3	Depends on JP25,JP27,JP28,JP29
JP14	Output's supply	PC 5V (no isolation)	External 5VF (isolation)	2-3	With JP15, depends on JP25,JP27,JP28,JP29
JP15	Output's common	PC GND (no isolation)	External RTNF (isolation)	2-3	With JP14, depends on JP25,JP27,JP28,JP29
JP16	Enable H/W reset via RS232 break	Installed - enable	Removed - Disable	R	
JP17	X command 0	PWM/Pulse,0-5V (DOWN)	Analog ±10V (UP)	2-3	Pulse for stepper drive
JP18	X command 1	PWM/Dir,0-5V (DOWN)	Analog ±10V (UP)	2-3	Dir for stepper drive
JP19	Y command 0	PWM/Pulse,0-5V (DOWN)	Analog ±10V (UP)	2-3	Pulse for stepper drive
JP20	Y command 1	PWM/Dir,0-5V (DOWN)	Analog ±10V (UP)	2-3	Dir for stepper drive

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Default R/IN	Remarks
JP21	Z command 0	PWM/Pulse,0-5V (DOWN)	Analog ±10V (UP)	2-3	Pulse for stepper drive
JP22	Z command 1	PWM/Dir,0-5V (DOWN)	Analog ±10V (UP)	2-3	Dir for stepper drive
JP23	T command 0	PWM/Pulse,0-5V (DOWN)	Analog ±10V (UP)	2-3	Pulse for stepper drive
JP24	T command 1	PWM/Dir,0-5V (DOWN)	Analog ±10V (UP)	2-3	Dir for stepper drive
JP25	User's ground source	Installed - PC ground. No isolation	Removed - RTNF external ground. Isolation	R	2-pin jumper with JP27,JP28,JP29
JP26	PPI reset source	Master µP HSO.1	Main board reset	2-3	For factory use only
JP27	User's 5V source	Installed - PC 5V. No isloation	Removed - External 5VF supply. Isolation.	R	2-pin jumper with JP25,JP28,JP29
JP28	+12V source	Installed: PC (+12V). No isolation	Removed: external +12VF. Isolation.	R	2-pin jumper with JP25,JP27,JP29
JP29	-12V source	Installed - PC (-12V). No isolation.	Removed: external (-12VF). Isolation.	R	2-pin jumper with JP25,JP27,JP28
JP30	Non-volatile power source	Installed - PC +12V.	Removed. On board source	R	Removed from the board
JP31	Reset control	Installed. Disable reset from PC.(SB214ND)	Removed. Enable reset from PC.(SB214PC)	R	2-pin jumper.For factory use only.
JP32	IRQ14 select	Installed. IRQ14 active		R	Interrupt to the PC. Only one IRQ can be selected, which is done by installing the appropriate jumper
JP33	IRQ13 select	Installed. IRQ13 active		R	
JP34	IRQ12 select	Installed. IRQ12 active		R	
JP35	IRQ11 select	Installed. IRQ11 active		R	
JP36	IRQ10 select	Installed. IRQ10 active		R	
JP37	IRQ3 select	Installed. IRQ3 active		R	
JP38	IRQ4 select	Installed. IRQ4 active		R	
JP39	IRQ5 select	Installed. IRQ5 active		R	
JP40	IRQ7 select	Installed. IRQ7 active		R	
JP41	IRQ2 select	Installed. IRQ2 active		R	

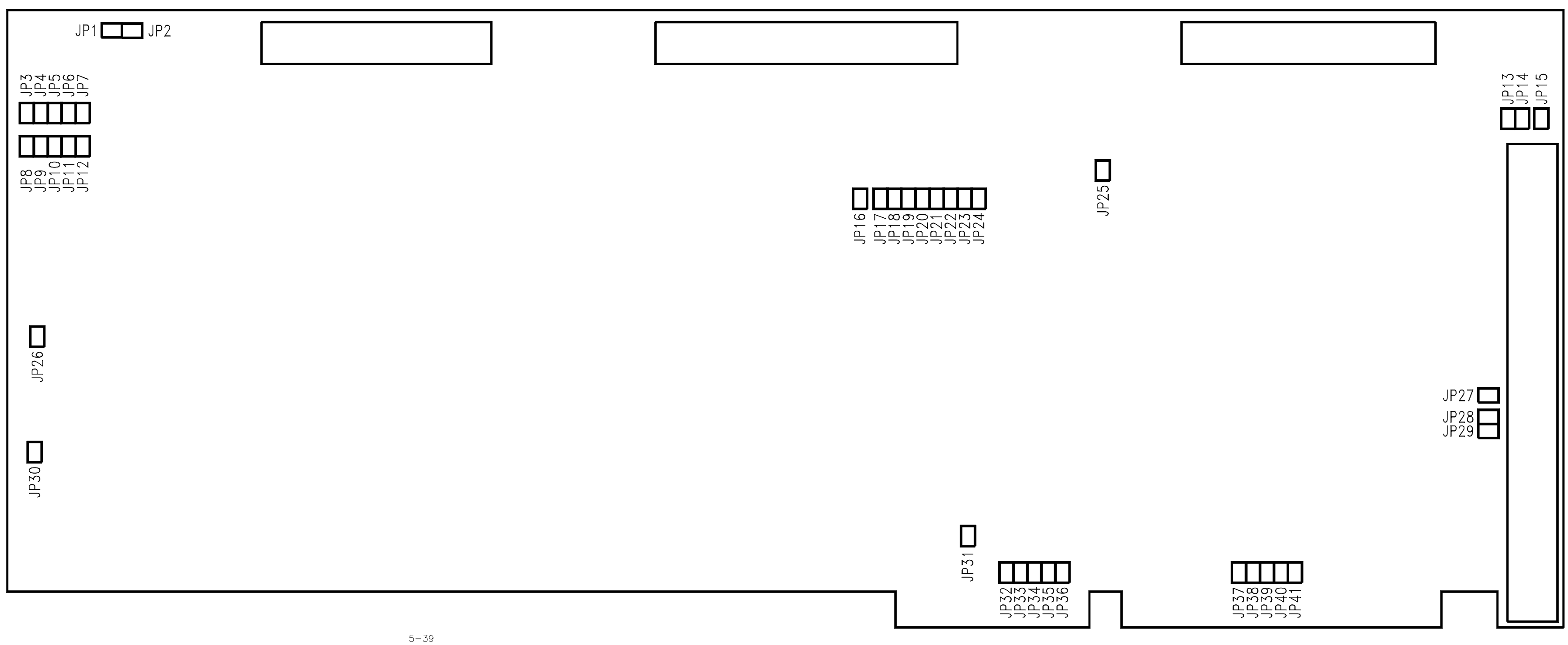

Figure 5-9 Jumpers SB214PC

5.1.19 External hardware Reset

The controller can be reset externally using RXD receive line. This can be done in two ways:

- Forcing a break command on the RS232 of the Host computer.
- Shorting the RS232 RXD to +12V using an external switch.

5.1.20 Address Setup

Using the Dip Switch, select the desired address. Address 000H to 3FFH in steps of 8 bytes can be selected. When using serial communication, all switches must be 'OFF'.

Switch	Function	100H	108H	2F8H
SW1	PC/RS232	ON	ON	ON
SW2	A3	ON	OFF	OFF
SW3	A4	ON	ON	OFF
SW4	A5	ON	ON	OFF
SW5	A6	ON	ON	OFF
SW6	A7	ON	ON	OFF
SW7	A8	OFF	OFF	ON
SW8	A9	ON	ON	OFF

5.1.21 Interrupt Setup

When using an Interrupt, one of 10 IRQs can be selected by installing one of the interrupt select jumpers JP32 through JP41.

See page 5-15.

5.1.22 SB214INT Distribution Box

The SB214INT is an optional distribution box. Its goal is to simplify the interfacing by using a dedicated connector for each function.

The SB214INT interfaces with the SB214PC via four ribbon cables as follows:

SB214INT	SB214PC
JA	J5
JB	J1
JC	J2
JD	J4

The **SB214INT** external (user) interface is similar to the **SB214ND** with the following exceptions:

- No AC supply connector.
- No internal supplies. 5VF, ±12VF, 24F must be supplied by the user.
- Driver box interface. The Command0, Command1 signals format is identical on both PWM0 and V0, and PWM1 and V1 pins. The selection of the format (analog or PWM) is done on the SB214PC by setting the appropriate jumpers.

5.2 SB214PC2 PC Based Dual Axis Controller

The SB214PC2 can be interfaced directly via J4 and J6 connectors, encoders, I/O, etc., using flat ribbon cables. All signals are buffered and/or isolated on board.
For simple interface the SB214PC2INT kit is offered. It consists of the following:

- One 26pin flat ribbon cable. 180cm long.
- One 60pin flat ribbon cable. 180cm long.
- One FLKM-26 Phoenix Varioface module. (Flat cable to screw terminal).
- One FLKM-60 Phoenix Varioface module. (Flat cable to screw terminal).

5.2.1 Power Supply

All drive interface and I/Os signals are opto- isolated. To maintain the isolation, an external supply must be used to supprt the isolated, user's side. The non isolated side is supplied by the PC supply via the PC bus.

Controller Non Isolated Supply

The PC Bus supply. These voltages appear in the Flat Cable Connector:

COMMON

+5V

+12V

-12V

+5VFU (+5V through 2.5A fuse)

User Isolated Supply

An external supply:

TNF (Field common)

+5VF

+12VF

-12VF

These voltages can be supplied from the PC (thus loosing the isolation) by using the jumpers JP5,6,7 and 15. (See page 5-31)

5.2.2 Drive Interface

The controller can control the following types of motors/drivers:

- A stepper motor driver with two phase current commands (XT=0).
- A servo motor with single torque/velocity command (XT=1).
- A stepper motor driver with Pulse-Dir command (XT=2).
- A sinusoidal brushless motor with a drive that accepts two phase current commands (the controller does the commutation) (XT=3).
- An AC induction motor with a drive that accepts two phase current commands, using Field Oriented Control (Vector Control) method (XT=5).

The interface consists of the following signals:

PWM0,PWM1 (or V0,V1), Enable, Fault, Settling, RTNF (return).

PWM0 (V0), PWM1 (V1) Servo Drive Commands

The command is available in two forms:

PWM	0-5V level. 50% duty cycle represent zero command.
Analog	±10V. It is generated by passing the PWM signal through a low pass filter.

The desired signal is selected by jumpers JP11 to JP14.
(See page 5-31)

Offset adjustment

The offset of V0, V1 can be adjusted using the following potentiometers:

Axis	Signal	Potentiometer
X	V0	RV3
X	V1	RV4
Y	V0	RV1
Y	V1	RV2

Pulse-Dir Drive Command

When using a driver that accepts Pulse-Dir input (step or servo motor), then the PWM0,1 designated commands must be used, PWM0 is the Pulse (clock) signal, and PWM1 is the Dir signal.
The pulse width is programmable by the Pulse Width ($PW) parameter in the range [0.4 - 60] microsecond.

The pulse rate range is [1 - 2,000,000] pulses/sec.

The aXis Type ($XT) must be set to 2. Then controller outputs the Pulse-Dir signals via the PWM0,1 designated outputs.

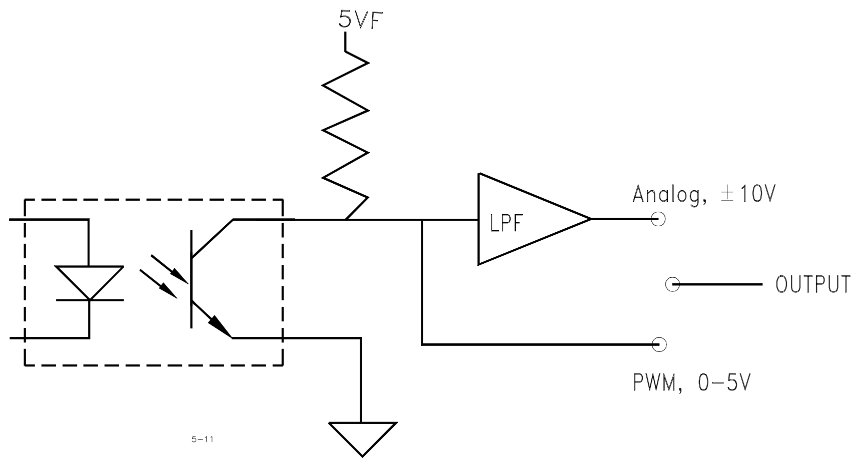

Figure 5-10 SB214PC2 PWM / analog command selection

JP17-JP24 select the type of command for a specific axis.

Enable (Output)

The enable signal polarity is programmable via the $OL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that are specified by the $Enable Deleay ($ED) before it checks the status of the drive fault feedback signal.

The signal is isolated. See Figure 5-11

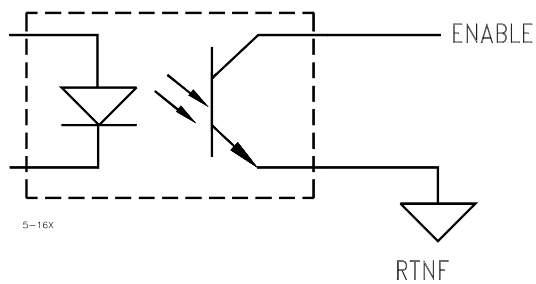

Figure 5-11 SB214PC2 Enable output

Fault (Input)

The drive fault signal polarity is programmable via the $IL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that are specified by the $Enable Deleay ($ED) before it checks the status of the drive fault feedback signal.

When the fault feedback becomes TRUE, an error message E26 is generated.

The signal is isolated. See Figure 5-12.

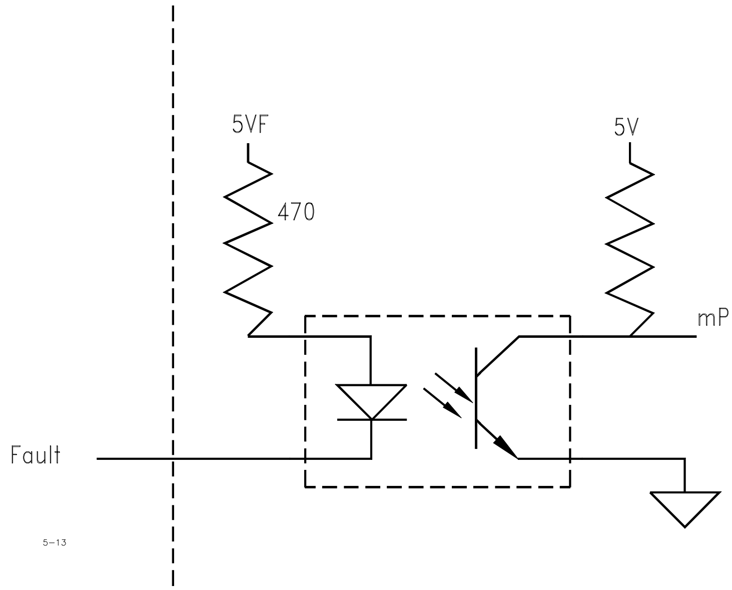

Figure 5-12 SB214PC2 drive fault input

Settling

This signal is used with the ST-4 micro-stepper drivers for four phase stepper motors. When enabled, it shorts two out of the four phases, thus producing a significant damping effect.

The electric circuit of the Settling Output is identical to that of the Enable output.

5.2.3 Encoder Interface

There are three encoder ports - encoder #1 (primary) for X,Y and one encoder #2 (secondary) for X.

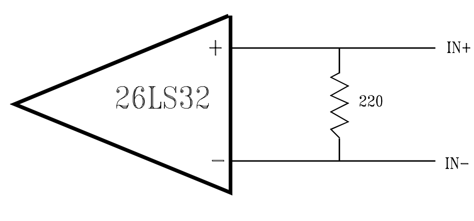

Figure 5-13 Encoder interface

Each encoder input port accepts three channel differential, TTL level, optical encoder, or laser interferometer signals of various formats. The format and the maximum counting speed is programmable via the Encoder Type (**ET** - primary, **et** - secondary) parameters.

- UP-DOWN plus index.
  The maximum counting rate is either 10 MHz ($ET=0), or 20 MHz ($ET=100).

- CLOCK-DIR plus index.
  The maximum counting rate is either 10 MHz ($ET=1), or 20 MHz ($ET=101).
- Two adjacent bits (C0, C1) of an up-down counter plus index.
  The maximum counting rate is either 10 MHz ($ET=2), or 20 MHz ($ET=102).
- Two phase quadrature plus index. (Factory default, 10MHz).
  The maximum counting rate (including X4) is either 10 MHz ($ET=3), or 20 MHz ($ET=103).

The input buffer is built around AM26LS32 line receivers. It is recommended to use encoders with built-in line drivers (AM26LS31 or similar).

The encoder can be powered by the PC 5V via the +5VFU (fused) pin, or externally by an external supplied source. When an external supply is used, its return must be connected to the (PC) non-isolated common.

CHA+	Encoder A non-inverted input (CLOCK,UP,C0)
CHA-	Encoder A inverted input (CLOCK,UP,C0)
CHB+	Encoder B non-inverted input (DIR,DOWN,C1)
CHB-	Encoder B inverted input (DIR,DOWN,C1)
CHI+	Encoder I non-inverted input
CHI-	Encoder I inverted input

5.2.4 Input Port

There are 16 general purpose inputs IN1 - IN16. All inputs are opto-coupled. 10MHz opto-couplers are used for the registration inputs (6,7).
See Figure 5-14.

The state of the input is interpreted by the controller based on the input level and on the corresponding IN (INput logic) bit. When the IN bit is 0, then the input is "on" when the input pin is shorted to RTNF.
The registration inputs activate the AUTO_M1 routines and cause latching of the mark position when shorting the input to RTNF independent of the IN bit setting.

IN1 to IN5 are supported by ACSPL automatic routines (AUTO_IN1 - AUTO_IN5).

The **ACSPL IP** parameter holds the binary representation of the 16 inputs status.

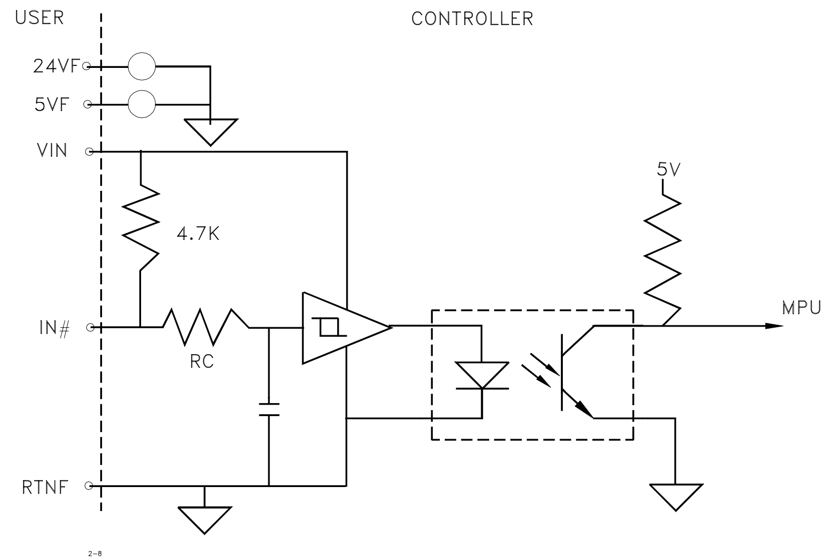

Figure 5-14 SB214PC2 Input port interface

The fast registration inputs (6,7) filter (RC) has a time constant of 1 µsec. The rest of the inputs, including the safety inputs, have a time constant of 1 msec.

5.2.5 Safety Inputs

The EMERGENCY STOP input is used to stop and deactivate the axis instantaneously.
The limit inputs are used to avoid axis over travel by preventing further motion in the inhibited direction.
The structure of the safety inputs is identical to the structure of the general purpose inputs.
The polarity of the safety inputs can be altered by the Input Logic (IL) parameter.

5.2.6 Outputs

There are 16 general purpose outputs. Outputs may be programmed to serve a dedicated function. See **Input & Output Ports**, page 7-1 for full description.

All outputs are opto-isolated with open collector. Outputs 1,2,3,8 use fast 10MHz opto-couplers and can sink 10mA. The output level must be limited to 0-5V range. Outputs 4,5,6,7,9,10,11,12,13,14,15,16 use "slow" opto-couplers (10 - 50 kHz) and can sink 2.5mA. The output level must be limited to 0-24V range. See Figure 5-15.

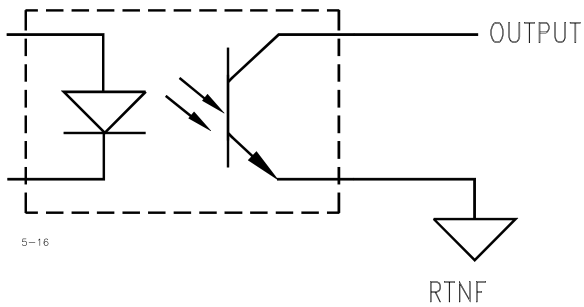

Figure 5-15 SB214PC2 Output port

5.2.7 Analog / Joystick Inputs

There are two Analog Inputs - AI1 and AI2. Analog Inputs 1 (ACSPL's A0) and 2 (ACSPL's A1) can be used as joystick inputs. AI1 for X and Z, and AI2 for Y and T.

The joystick potentiometer must be connected between the 5FU and its return. See Figure 5-16. The potentiometer output is connected to AI1 and AI2 pins.

The two inputs act also as general purpose analog inputs that are sampled via a 10 bits ADC by the A0 and A1 ACSPL variables. 0V is represented by 0, and 5V is represented by 1024. The user must assure that the input voltage does not exceed the specified range of [0 - 5V].

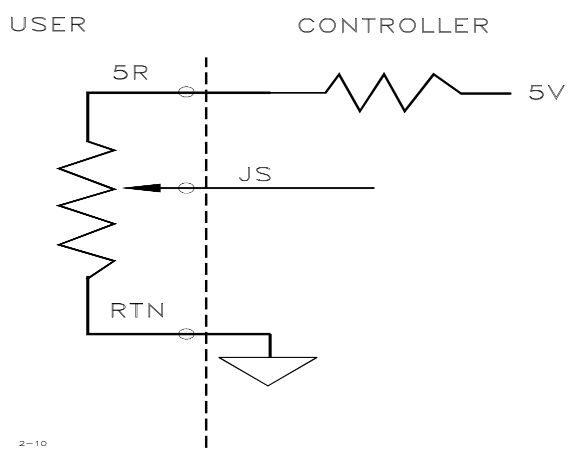

Figure 5-16 SB214PC2 Joystick connection

5.2.8 Analog Outputs

There are three Analog Outputs, in the range -10V - +10V.
The analog output (**A_OUT**) is an 8 bit general purpose that can be set via the **D0** parameter.
There is one 10 bits analog output for each axis, to be used for monitoring purposes, or as an uncommitted, general purpose analog output - **X_A (XD4), Y_A(YD4), Z_A(ZD4), T_A(TD4).**

To use an axis output as an uncommitted output, set $MN to 0. For example, to use XD4 - **SXMN0<cr>.**

To use an axis output to monitor a real time signal, set $MN to 1 (velocity), 2 (current command), or 3. When $MN=3, $DC points to the required parameter. For example, to monitor the position error of Y axis - **SYMN3 DC4<cr>**

Name	Description
COM	Common (ground)
X_A	X Analog Out - XD4
Y_A	Y Analog Out - YD4
A_O	General purpose Analog Output - D0

5.2.9 Serial Communication

Asynchronous serial communication to host.

To communicate exclusively through the serial link, jumper JP23 must be installed. SW1 to SW8 dip switches must be in 'OFF' position.

While communicating through the PC bus, the controller also listens to the serial link. When a SFM<cr> command is received, the controller switches to communication through the serial link only.

RXD	RS232 receive.
TXD	RS232 transmit.

5.2.10 J3 Supply Connector

Type: Molex 5566-4. Mating: Molex 5557-4

When the controller is used outside a PC, it must be powered by an external 5V, +/-12V supply.

Pin	Name	Description
1	+12V	12V supply. 0.1A consumption.
2	-12V	-12V supply. 0.1A consumption.
3	Common	5, +/-12V common.
4	5V	5V supply. 1.5A consumption w/o external loads (encoders).

5.2.11 J4 Connector

26 pins flat ribbon cable

Pin	Name	Description
1	TXD	RS232 TXD
2	RXD	RS232 RXD
3	M_SD	For factory use only.
4	Common	PC common for 5V, ±12V
5	+12V	PC +12V
6	-12V	PC -12V
7	M_LED	μP ON LED command (TTL)
8	Y_A	Y analog out
9	IN9	Input 9
10	IN10	Input 10
11	IN11	Input 11
12	IN12	Input 12
13	IN13	Input 13
14	IN14	Input 14
15	IN15	Input 15
16	IN16	Input 16
17	OUT9	Output 9
18	OUT10	Output 10
19	OUT11	Output 11
20	OUT12	Output 12
21	OUT13	Output 13
22	OUT14	Output 14
23	OUT15	Output 15
24	OUT16	Output 16
25	N.C.	For future use
26	N.C.	For future use

5.2.12 J6 Connector

60 pins flat ribbon cable

Pin	Name	Description
1	+5VFU	5V fused
2	Common	PC common for 5V, ±12V
3	X_EN	X Enable
4	X_FLT	X Fault input
5	X_Command0	X PWM0/Command0
6	X_Command1	X PWM1/Command1
7	X_RL	X Right Limit
8	X_LL	X Left Limit

Pin	Name	Description
9	X_CH1I+	X Encoder #1 Index+
10	X_CH1I-	X Encoder #1 Index-
11	X_CH1A+	X Encoder #1 A+
12	X_CH1A-	X Encoder #1 A-
13	X_CH1B+	X Encoder #1 B+
14	X_CH1B-	X Encoder #1 B-
15	Y_EN	X Enable
16	Y_FLT	X Fault input
17	Y_Command0	Y PWM0/Command0
18	Y_Command1	Y PWM1/Command1
19	Y_RL	Y Right Limit
20	Y_LL	Y Left Limit
21	Y_CH1I+	Y Encoder #1 Index+
22	Y_CH1I-	Y Encoder #1 Index-
23	Y_CH1A+	Y Encoder #1 A+
24	Y_CH1A-	Y Encoder #1 A-
25	Y_CH1B+	Y Encoder #1 B+
26	Y_CH1B-	Y Encoder #1 B-
27	X_A	X analog out
28	RTNF	5VF return
29	VIN	Field 5-24V for the Inputs, Limits and Emergency Stop.
30	5VF	Field 5V. Must be supplied externally
31	+12VF	Field 12V. Must be supplied externally
32	-12VF	Field -12V. Must be supplied externally
33	IN1	Input 1
34	IN2	Input 2
35	IN3	Input 3
36	IN4	Input 4
37	IN5	Input 5
38	IN6	Input 6
39	IN7	Input 7
40	IN8	Input 8
41	OUT1	Output 1
42	OUT2	Output 2
43	OUT3	Output 3
44	OUT4	Output 4
45	OUT5	Output 5
46	OUT6	Output 6
47	OUT7	Output 7
48	OUT8	Output 8
49	AI1	Analog input 1 (Joystick)

Pin	Name	Description
50	AI2	Analog input 2 (Joystick)
51	X_CH2I+	X encoder #2 index+
52	X_CH2I-	X encoder #2 index-
53	X_CH2A+	X encoder #2 A+
54	X_CH2A-	X encoder #2 A-
55	X_CH2B+	X encoder #2 B+
56	X_CH2B-	X encoder #2 B-
57	ES	Emergency Stop
58	A_OUT	General purpose analog out
59	+5VFU	5V fused
60	Common	PC common for 5V, ±12V

5.2.13 SB214PC2 - Jumpers

See Figure 5-17.

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Def. R/IN	Remarks
JP1	PPI intrpt source	Installed - Master µP HSO.0	Removed - SB2500	R	For factory use only
JP2	H/W watchdog	Disabled	Enabled	2-3	For factory use only
JP3	Master µP clock source	installed - PC system clock	removed - dedicated clock	IN	2 pins jumper, For factory use only
JP4	Out3 (J3 pin 43) select	Out3	Equal to Out16	1-2	
JP5	+12V source	Installed: PC (+12V). No isolation	Removed: external +12VF. Isolation.	R	2 pins jumper. To be set with JP6,JP7,JP15
JP6	-12V source	Installed - PC (-12V). No isolation.	Removed: external (-12VF). Isolation.	R	2 pins jumper. To be set with JP5,JP7,JP15
JP7	User's ground source	Installed - PC ground. No isolation	Removed - RTNF external ground. Isolation	R	2 pins jumper. To be set with JP5,JP6,JP15
JP8	Inputs, limits, ES common	External RTNF (isolation)	PC GND (no isolation)	1-2	Depends on JP5,JP6,JP7,JP15
JP9	Output's common	external RTNF (isolation)	PC GND (no isolation)	1-2	To be set with JP10. Depends on JP5,JP6,JP7,JP15
JP10	Output's supply	External 5VF (isolation)	PC 5V (no isolation)	1-2	To be set with JP9. Depends on JP5,JP6,JP7,JP15
JP11	X command 0	PWM/Pulse,0-5V	Analog ±10V	2-3	Pulse for stepper drive
JP12	X command 1	PWM/Dir,0-5V	Analog ±10V	2-3	Dir for stepper drive
JP13	Y command 0	PWM/Pulse,0-5V	Analog ±10V	2-3	Pulse for stepper drive
JP14	Y command 1	PWM/Dir,0-5V	Analog ±10V	2-3	Dir for stepper drive
JP15	User's 5V source	Installed - PC 5V. No isloation	Removed - External 5VF supply. Isolation.	R	2 pins jumper. To be set with JP5,JP6,JP7
JP16-18	Future use			R	
JP19	OUT6/STLX	XSTLN	OUT6	2-3	
JP20	OUT7/STLY	YSTLN	OUT7	2-3	
JP21	Enable H/W reset via RS232 break	Installed - enable	Removed - Disable	R	
JP22	Non-volatile power source	Installed - PC +12V.	Removed. On board source	R	2 pins jumper.For factory use only.

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Def. R/IN	Remarks
JP23	Reset control	Installed. Disable reset from PC.(SB214ND)	Removed. Enable reset from PC.(SB214PC)	R	2 pins jumper.Install when using as stand alone..
JP24	IRQ2 select	Installed. IRQ2 active		R	
JP25	IRQ5 select	Installed. IRQ5 active		R	
JP26	IRQ4 select	Installed. IRQ4 active		R	
JP27	IRQ3 select	Installed. IRQ3 active		R	
JP28	IRQ7 select	Installed. IRQ7 active		R	

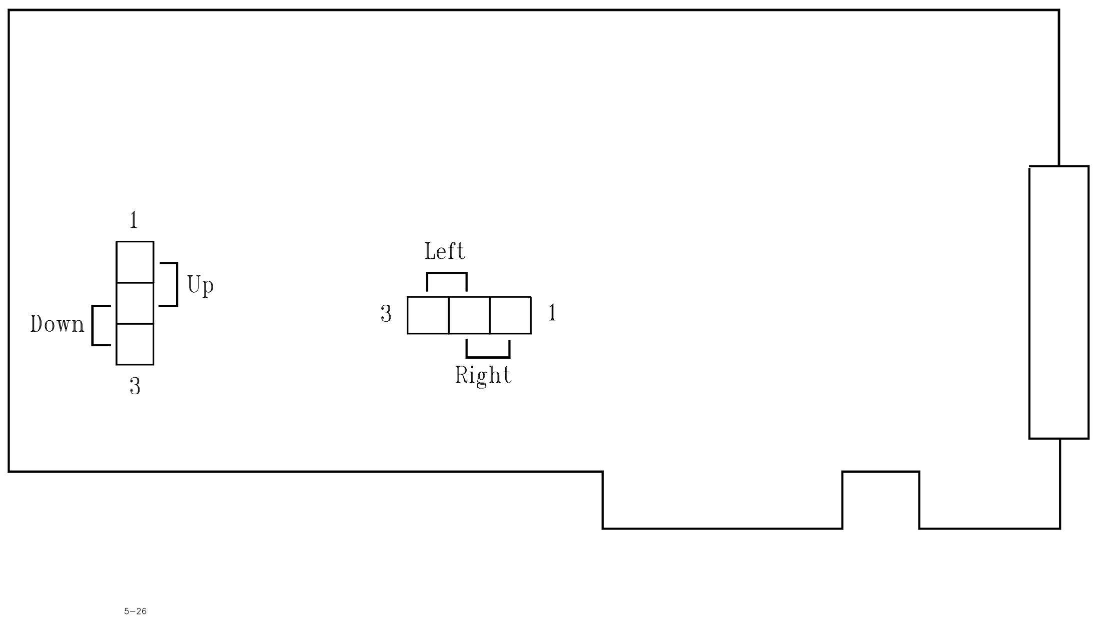

Figure 5-17   SB 214 PC2 jumpers definition

5.2.14 Address Setup

Using the Dip Switch, select the desired address. Address 000H to 3FFH in steps of 8 bytes can be selected. When using serial communication, all switches must be 'OFF'.

Switch	Function	100H	108H	2F8H
SW1	PC/RS232	ON	ON	ON
SW2	A3	ON	OFF	OFF
SW3	A4	ON	ON	OFF
SW4	A5	ON	ON	OFF
SW5	A6	ON	ON	OFF
SW6	A7	ON	ON	OFF
SW7	A8	OFF	OFF	ON
SW8	A9	ON	ON	OFF

SW1 is used to select the type of communication. parallel PC bus or serial link. When communicating with the PC via the parallel bus make sure that SW1 is in the ON position.

The address of the unit in the I/O map of the PC is set by SW8-SW2 (A9-A3).

It can be set in banks of 8 bytes, from address 000H to 3F8H.

Within a bank, the address offsets are:

PC's Receive (Controller'sTransmit) FIFO - 0 (read)

PC's Transmit (Controller's Receive) FIFO - 0 (write)

Communication flags register - 1 (read only)

5.2.15 External Hardware Reset

This option enables the user to reset the board externally, using the RXD receive line. It can be done in one of the following ways:

- Forcing a break command on the RS232 of the Host computer.
- Short the RS232 RXD to +12V using an external switch.
  This option is enabled by installing JP16.

(To reset the controller via the communication link use the HWRES command. It has the same effect as turning the power off and on. RESET command also resets the controller. However, it also initializes all parameters to factory default.)

5.2.16 Driver Interface

Commands 0,1 of each axis can be set to a ±10V analog command, or 0-5V PWM type command. The PWM command is used for drivers with PWM input or Pulse and Direction input.

JP11-14 select the type of command for a specific axis.

5.2.17 Isolation And External Supplies

The drivers, the input port and the output port can be isolated from the controller/PC by setting the appropriate jumpers and using external isolated supplies. The following matrix descibes the relation between the setting of the jumpers and the various isolations. Note that if there is no isolation between two isolated parties, for example, if the input port and the drive control are not isolated, then there is no isolation between the drive and the input.

Drivers	Input port	Output port	JP5	JP6	JP7	JP15	JP8	JP9	JP10
IS	IS	IS	OFF	OFF	OFF	OFF	1-2	1-2	1-2
IS	IS	NI	OFF	OFF	OFF	OFF	1-2	2-3	2-3
IS	NI	IS	OFF	OFF	OFF	OFF	2-3	1-2	1-2
IS	NI	NI	OFF	OFF	OFF	OFF	2-3	2-3	2-3
NI	NI	NI	ON	ON	ON	ON	2-3	2-3	2-3

IS - Isolated

NI - No Isolation

Interrupt Setup

When using an Interrupt, one of 5 IRQs can be selected, by installing one of the interrupt select jumpers JP24 through JP28.

5.3 SB214VME VME Based 4 Axis Controller

The SB214VME has four connectors. One (P1, rear upper) is used for interfacing with the VME bus. The other three are for external interfacing.

Some of the signals are available via the front connectors only, some via the P2 rear connector only, and some via both.

The following table lists which functions are available via the front and back connectors:

Function	Front Connectors	Rear Connector (P2)
Controller supply (GND,+5V)	Present	-
Field Supply (RTNF,+5VF,±12VF)	Present	Present ((*))
X,Y,Z,T Driver interface	Present	Present ((*))
X,Y,Z,T Limits	Present	Present ((*))
Emergency Stop	Present	Present ((*))
In 1 - 8	Present	Present
In 9 -16	Present	-
Out 1 -8	Present	Present ((*))
Out 9 -16	Present	-
X,Y,Z,T Encoder #1	Present	-
X,Y Serial interface	-	Present
T Serial interface	-	Present
Z,T Encoder #2	-	Present
X,Y,Z,T Analog out	Present	Present ((*))
General purpose Analog Out	Present	Present ((*))
Joystick Analog In1,2	Present	Present ((*))
Host Serial Communication	-	Present
Extension board interface	-	Present

(*) Present only in the 96 pins P2.

5.3.1 Power Supply

All drive interface and I/Os signals are opto- isolated. To maintain the isolation, an external supply must be used to supprt the isolated, user’s side. The non islated side is supplied by the VME supply via the VME bus.

Controller Non Isolated Supply

Supply from the VME Bus:

- COMMON
- +5V
- +12V
- -12V

User Isolated Supply

An external supply:

RTNF (Field common)

+5VF

+12VF

-12VF

These voltages can be supplied from the VME (thus loosing the isolation) by using the jumpers JP19, JP20, JP21. (See page 5-53)

Drive Interface

The controller can control the following types of motors/drivers:

- A stepper motor driver with two phase current commands. (XT=0)
- A servo motor with single torque/velocity command. (XT=1)
- A stepper motor driver with Pulse-Dir command. (XT=2)
- A sinusoidal brushless motor with drive that accepts two phase current commands (the controller does the commutation). (XT=3)
- An AC induction motor with drive that accepts two phase current commands, using Field Oriented Control (Vector Control) method. (XT=5)

The interface consists of the following signals:

PWM0,PWM1 (or V0,V1), Enable, Fault, Settling, RTNF (return).

PWM0 (V0), PWM1 Servo Drive Commands

The command is available in two forms:

PWM	0-5V level. 50% duty cycle represent zero command.
Analog	±10V. It is generated by passing the PWM signal through a low pass filter. Only V0 is available in this form

The desired signal is selected by jumpers JP17 to JP24.
(See page 5-53)

Offset adjustment

The offset of V0 can be adjusted using the following potentiometers:

Axis	Signal	Potentiometer
X	V0	RV1
Y	V0	RV2
Z	V0	RV3
T	V0	RV4

Pulse-Dir Drive Command

When using a driver that accepts Pulse-Dir input (step or servo motor), then the PWM0,1 designated commands must be used, PWM0 is the Pulse (clock) signal, and PWM1 is the Dir signal.
The pulse width is programmable by the Pulse Width ($PW) parameter in the range [0.4 - 60,000] microsecond.The pulse rate range is [1 - 2,000,000] pulses/sec.
The aXis Type ($XT) must be set to 2. Then controller outputs the Pulse-Dir signals via the PWM0,1 designated outputs.

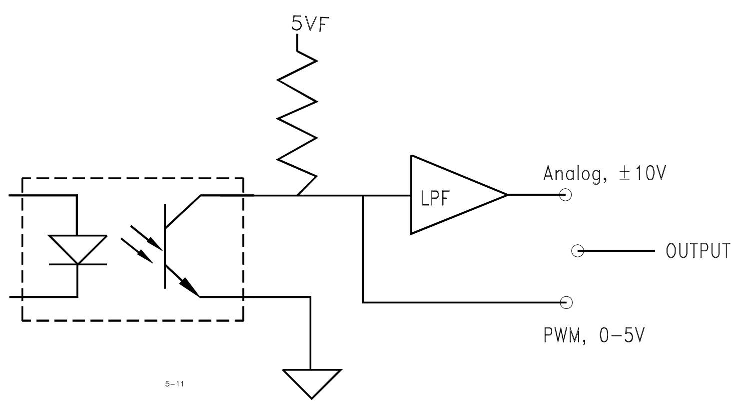

Figure 5-18 SB214VME PWM/analog selection

JP17-JP24 select the type of command for a specific axis.

Enable (Output)

The enable signal polarity is programmable via the $OL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that is specified by the Enable Delay ($ED) parameter before it checks the status of the drive fault feedback signal.

The signal is isolated. See Figure 5-19.

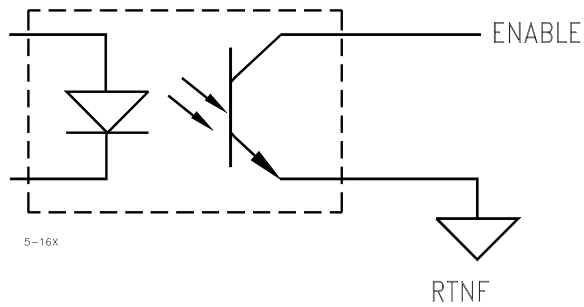

Figure 5-19   SB214VME Srive enable output

Fault (Input)

The drive fault signal polarity is programmable via the $IL parameter (See Software Guide).

Once the driver is enabled, the controller waits the number of milliseconds that are specified by the $Enable Deleay ($ED) before it checks the status of the drive fault feedback signal.

When the fault feedback becomes TRUE, an error message E26 is generated.

The signal is isolated. See Figure 5-20.

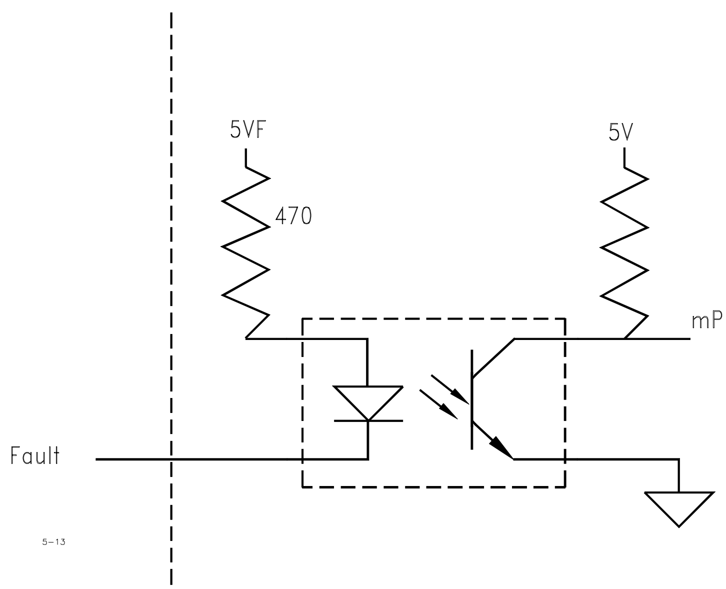

Figure 5-20   SB214VME drive fault

Settling

This signal is used with the DR170 micro-stepper drivers for four phase stepper motors. When enabled, it shorts two out of the four phases, thus producing a significant damping effect.

The electric circuit of the Settling Output is identical to that of the Enable output.

5.3.2 Encoders

There are six encoder ports - encoder #1 (primary) for X,Y,Z and T, and encoder #2 (secondary) for Z and T. Each can serve a different function according to the setup of the specific axis.

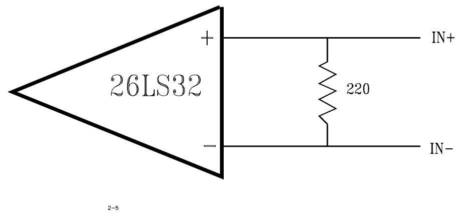

Figure 5-21 Encoder interface

Each encoder input port accepts three channel differential, TTL level, optical encoder, or laser interferometer signals of various formats. The format and the maximum counting speed is programmable via the Encoder Type (**ET** - primary, **et** - secondary) parameters.

- UP-DOWN plus index.
  The maximum counting rate is either 10 MHz ($ET=0), or 20 MHz ($ET=100).
- CLOCK-DIR plus index.
  The maximum counting rate is either 10 MHz ($ET=1), or 20 MHz ($ET=101).
- Two adjacent bits (C0, C1) of an up-down counter plus index.
  The maximum counting rate is either 10 MHz ($ET=2), or 20 MHz ($ET=102).
- Two phase quadrature plus index. (Factory default, 10MHz).
  The maximum counting rate (including X4) is either 10 MHz ($ET=3), or 20 MHz ($ET=103).

The input buffer is built around AM26LS32 line receivers. It is recommended to use encoders with built-in line drivers (AM26LS31 or similar).

The encoder can be powered by the PC 5V via the +5VFU (fused) pin, or externally by an external supplied source. When an external supply is used, its return must be connected to the (PC) non-isolated common.

CHA+	Encoder A non-inverted input (CLOCK,UP,C0)
CHA-	Encoder A inverted input (CLOCK,UP,C0)
CHB+	Encoder B non-inverted input (DIR,DOWN,C1)
CHB-	Encoder B inverted input (DIR,DOWN,C1)
CHI+	Encoder I non-inverted input
CHI-	Encoder I inverted input

5.3.3 Input Port

There are 16 general purpose inputs. Some inputs may serve dedicated functions. See **Input & Output Ports**, page 7-1. All are opto-isolated. See Figure 5-22. The registration mark inputs (6,7,9,10) use fast, 10MHz opto-couplers.

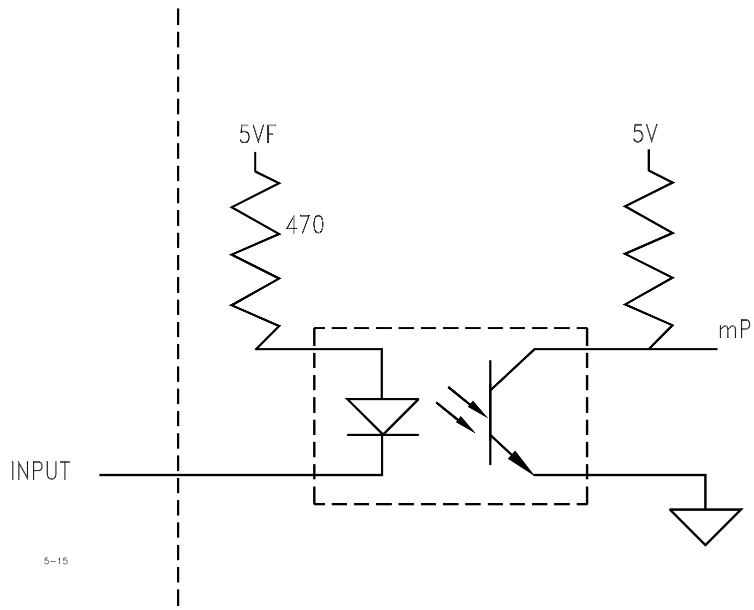

Figure 5-22   SB214VME input port

5.3.4 Safety Inputs

The EMERGENCY STOP input is used to stop and deactivate the axis instantaneously.

The limit inputs are used to avoid axis over travel by preventing further motion in the inhibited direction.

The structure of the safety inputs is identical to the structure of the general purpose inputs.

The polarity of the safety inputs can be altered by the Input Logic (IL) parameter.

5.3.5 Output Port

There are 16 general purpose outputs. Outputs can be programmed to serve a dedicated function. See **Input & Output Ports**, page 7-1.

All outputs are opto-isolated with open collector. See Figure 5-23.

Outputs 1,2,3,8,9,10,11,16 use fast 10MHz opto-couplers. Each of these outputs can sink 10mA. The output level must be limited to 0 - 5V range.

Outputs 4,5,6,7,12,13,14,15 use slow opto-couplers (10-50khz). Each of these outputs can sink at least 2.5mA. The output level must be limited to 0-24V range.

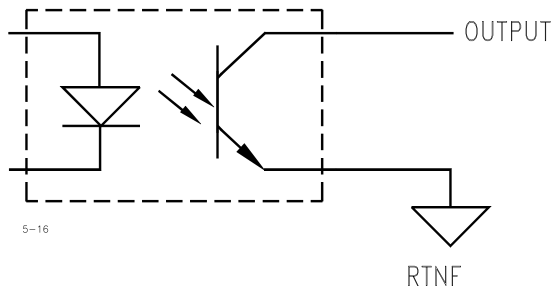

Figure 5-23 SB214VME output port

5.3.6 High Speed Serial Interface ("Interferometer")

There are three (X, Y and T axes), general purpose high speed synchronous serial interface ports. They can be used for high speed laser interferometers position reading, A to D interface, or any other function as required by the user. All signals are TTL level and must be buffered properly when used. Consult factory if you need to use them.

Name	Description
$_ADC_DI	Serial data input.
$_CONVST	Convert start signal.
$_SCLK	Serial clock.
ADCCH0,1,2	Channel select ID number. Common to all axes.
$_CH2I_DS	Interferometer Disconnected feedback. X, Y only.

5.3.7 Analog / Joystick Inputs

There are two Analog Inputs - AI1 and AI2. Analog Inputs 1 (ACSPL's A0) and 2 (ACSPL's A1) can be used as joystick inputs. AI1 for X and Z, and AI2 for Y and T.

The joystick potentiometer must be connected between the 5V and its return. See Figure 5-24.

The potentiometer output is connected to AI1 and AI2 pins.
The two inputs act also as general purpose analog inputs that are sampled via a 10 bits ADC by the A0 and A1 ACSPL variables. 0V is represented by 0, and 5V is represented by 1024. The user must assure that the input voltage does not exceed the specified range of [0 - 5V].

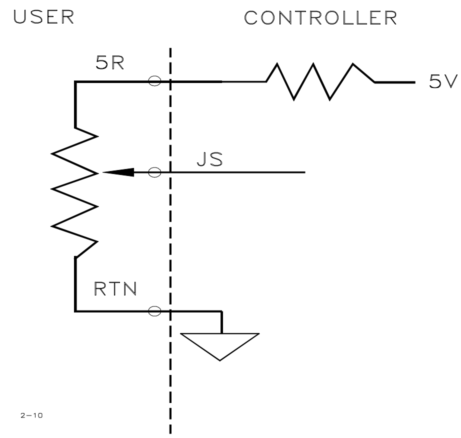

Figure 5-24 Joystick connection

Name	Description
AI2	Analog Input 2 ACSPL - A1
AI1	Analog Input 1 ASPL - A0
COM	5V return

5.3.8 Analog outputs

There are five Analog Outputs, in the range -10V - +10V.
The analog output (**A_OUT**) is an 8 bit general purpose that can be set via the **D0** parameter.
There is one 10 bits analog output for each axis, to be used for monitoring purposes, or as an uncommitted, general purpose analog output - **X_A (XD4), Y_A(YD4), Z_A(ZD4), T_A(TD4).**

To use an axis output as an uncommitted output, set $MN to 0. For example, to use XD4 - **SXMN0<cr>.**

To use an axis output to monitor a real time signal, set $MN to 1 (velocity), 2 (current command), or 3. When $MN=3, $DC points to the required parameter. For example, to monitor the position error of Y axis - **SYMN3 DC4<cr>**

Name	Description
COM	Common (ground)
X_A	X Analog Out - XD4
Y_A	Y Analog Out - YD4
Z_A	Z Analog Out - ZD4
T_A	T Analog Out - TD4
A_O	General purpose Analog Output - D0

5.3.9 Serial Communication

Asynchronous serial communication to host.

To communicate exclusively through the serial link, jumper JP31 must be installed. SW1 dip switch must be in ‘OFF’ position.

While communicating through the VME bus, the controller also listens to the serial link. When a SFM<cr> command is received, the controller switches to communication through the serial link only.

RXD	RS232 receive.
TXD	RS232 transmit.
M_SD	Communication shutdown. TTL level.

5.3.10 Extension Serial Interface

Interface to extension modules.

S_RXDI	Receive. TTL level.
S_TXDO	Transmit. TTL level.
S_SD	Shut down. TTL level
IO_S0 - IO_S4	Select lines. TTL level.

5.3.11 J1 Connector (Front Lower)

Type: 60 pins flat cable.

J1 includes the following types of signals:

T axis limit switch inputs	T_LL, T_RL
General purpose outputs	OUT1..OUT16
General purpose inputs	IN1..IN16
Drive’s interface	$_EN, $_FLT, $_PWM0/1, $_STL
Field supply	+5VF, +12VF, -12VF, RTNF

Pin	Signal	I/O	Description
1	5VF	Sup	Field 5V
2	RTNF	Sup	Field Return
3	+12VF	Sup	Field +12V
4	-12VF	Sup	Field -12V
5	X_EN	Out	X drive Enable
6	X_FLT	In	X drive Fault feedback
7	X_PWM0	Out	X PWM0 drive command
8	X_PWM1	Out	X PWM1 drive command
9	X_STL	Out	X Settling command. (For four phase steppers only)
10	Y_STL	Out	Y Settling command. (For four phase steppers only)

Pin	Signal	I/O	Description
11	Y_EN	Out	Y drive Enable
12	Y_FLT	In	Y drive Fault feedback
13	Y_PWM0	Out	Y PWM0 drive command
14	Y_PWM1	Out	Y PWM1 drive command
15	Z_EN	Out	Z drive Enable
16	Z_FLT	In	Z drive Fault feedback
17	Z_PWM0	Out	Z PWM0 drive command
18	Z_PWM1	Out	Z PWM1 drive command
19	Z_STL	Out	Z Settling command. (For four phase steppers only)
20	T_STL	Out	T Settling command. (For four phase steppers only)
21	T_EN	Out	T drive Enable
22	T_FLT	In	T drive Fault feedback
23	T_PWM0	Out	T PWM0 drive command
24	T_PWM1	Out	T PWM1 drive command
25	OUT1	Out	Output #1
26	OUT2	Out	Output #2
27	OUT3	Out	Output #3
28	OUT4	Out	Output #4
29	OUT5	Out	Output #5
30	OUT6	Out	Output #6
31	OUT7	Out	Output #7
32	OUT8	Out	Output #8
33	OUT9	Out	Output #9
34	OUT10	Out	Output #10
35	OUT11	Out	Output #11
36	OUT12	Out	Output #12
37	OUT13	Out	Output #13
38	OUT14	Out	Output #14
39	OUT15	Out	Output #15
40	OUT16	Out	Output #16
41	IN1	In	Input #1
42	IN2	In	Input #2
43	IN3	In	Input #3
44	IN4	In	Input #4
45	IN5	In	Input #5
46	IN6	In	Input #6
47	IN7	In	Input #7
48	IN8	In	Input #8
49	IN9	In	Input #9
50	IN10	In	Input #10

Pin	Signal	I/O	Description
51	IN11	In	Input #11
52	IN12	In	Input #12
53	IN13	In	Input #13
54	IN14	In	Input #14
55	IN15	In	Input #15
56	IN16	In	Input #16
57	T_RL	In	T Right Limit
58	T_LL	In	T Left Limit
59	+5V	Sup	CPU +5V
60	GND	Sup	CPU Ground

5.3.12 J2 Connector (Front Upper)

Type: 40 pins flat cable.

J2 includes the following types of signals:

Analog outputs	A_OUT(A_O), X_A, Y_A, Z_A, T_A
Joystick/Analog inputs	AI1, AI2
Emergency stop	ES
X, Y, Z limit inputs	$_LL, $_RL
Encoder interface	$_CH1A, $_CH1AN, $_CH1B, $_CH1BN, $_CH1I, $_CH1IN

Pin	Signal	I/O	Description
1	X_RL	In	X Right Limit
2	X_LL	In	X Left Limit
3	Y_RL	In	Y Right Limit
4	Y_LL	In	Y Left Limit
5	Z_RL	In	Z Right Limit
6	Z_LL	In	Z Left Limit
7	ES	In	Emergency Stop
8	GND	Sup	CPU Ground
9	Z_A	Out	Z Analog output. ±10V
10	T_A	Out	T Analog output. ±10V
11	X_A	Out	X Analog output. ±10V
12	Y_A	Out	X Analog output. ±10V
13	X_CH1I	In	X Encoder #1 Index+ (non inverted)
14	X_CH1IN	In	X Encoder #1 Index- (inverted)
15	X_CH1A	In	X Encoder #1 A+ (non inverted)
16	X_CH1AN	In	X Encoder #1 A- (inverted)
17	X_CH1B	In	X Encoder #1 B+ (non inverted)
18	X_CH1BN	In	X Encoder #1 B- (inverted)

Pin	Signal	I/O	Description
19	Y_CH1I	In	Y Encoder #1 Index+ (non inverted)
20	Y_CH1IN	In	Y Encoder #1 Index- (inverted)
21	Y_CH1A	In	Y Encoder #1 A+ (non inverted)
22	Y_CH1AN	In	Y Encoder #1 A- (inverted)
23	Y_CH1B	In	Y Encoder #1 B+ (non inverted)
24	Y_CH1BN	In	Y Encoder #1 B- (inverted)
25	Z_CH1I	In	Z Encoder #1 Index + (non inverted)
26	Z_CH1IN	In	Z Encoder #1 Index - (inverted)
27	Z_CH1A	In	Z Encoder #1 A+ (non inverted)
28	Z_CH1AN	In	Z Encoder #1 A- (inverted)
29	Z_CH1B	In	Z Encoder #1 B+ (non inverted)
30	Z_CH1BN	In	Z Encoder #1 B- (inverted)
31	T_CH1I	In	T Encoder #1 Index+ (non inverted)
32	T_CH1IN	In	T Encoder #1 Index- (inverted)
33	T_CH1A	In	T Encoder #1 A+ (non inverted)
34	T_CH1AN	In	T Encoder #1 A- (inverted)
35	T_CH1B	In	T Encoder #1 B+ (non inverted)
36	T_CH1BN	In	T Encoder #1 B- (inverted)
37	AI1	In	Analog in #1 (Joystick). 0 - 5V
38	AI2	In	Analog in #2 (Joystick). 0 -5V
39	A_OUT	Out	General pupose analog output
40	+5V	Sup	CPU +5V

5.3.13 J3(P2) Connector (Rear Lower Connector)

serial communication	M_RXD, M_TXD, M_SD
Extension serial interface	S_RXD, S_TXD, S_SD, IO_S0..IO_S4
Analog outputs	A_OUT (A_O), X_A, Y_A, Z_A, T_A
Analog inputs	AI1, AI2
High Speed Serial Interface (Interferometer)	T_ADC_DI, T_CONVST, T_SCLK, T_ADCCH0, T_ADCCH1, T_ADCCH2 XADC_DI, ACONVST, XSCLK, XCH2I_DS, XADC_DI, ACONVST, XSCLK, XCH2I_DS
Emergency stop	ES
Limit inputs	$_LL, $_RL
8 outputs	OUT1 .. OUT8
8 inputs	IN1..IN8
Drive's interface	$_EN, $_FLT, $_PWM0/1 or V0, V1, $_STL
Field supply	5VF, +12VF, -12VF, RTNF
Z, T Secondary encoder	Z_CH2I, Z_CH2A, Z_CH2B, Z_CH2I_DS, T_CH2I, T_CH2A, T_CH2B, T_CH2I_DS

Pin	Signal	I/O	Description
1A	M_RXD	In	Serial host communication RXD
2A	M_TXD	Out	Serial host communication TXD
3A	M_SD	Out	Serial host communication Shut Down
4A	S_RXD	I/O	Extension Board RXD
5A	S_TXD	Out	Extension Board TXD
6A	S_SD	Out	Extension Board Shut Down/Direction
7A	IO_S0	Out	TBD
8A	IO_S1	Out	TBD
9A	IO_S2	Out	TBD
10A	IO_S3	Out	TBD
11A	IO_S4	Out	TBD
12A	T_ADC_DI	In	T serial interface Data In
13A	T_CONVST	Out	T serial interface Start
14A	T_SCLK	Out	T serial interface Clock
15A	T_ADCCH0	Out	T serial interface Channel ID#0
16A	T_ADCCH1	Out	T serial interface Channel ID#1
17A	T_ADCCH2	Out	T serial interface Channel ID#2
18A	N.C.		
19A	T_LL	In	T Left Limit
20A	T_RL	In	T Right Limit

Pin	Signal	I/O	Description
21A	Z_LL	In	Z Left Limit
22A	Z_RL	In	Z Right Limit
23A	Y_LL	In	Y Left Limit
24A	Y_RL	In	Y Right Limit
25A	Y_STL	In	Y Settling command. (For four phase steppers only)
26A	Y_FLT	In	Y Fault input
27A	Y_PWM0	Out	Y PWM0 drive command
28A	Y_PWM1	Out	Y PWM1 drive command
29A	Y_EN	Out	Y drive Enable
30A	X_PWM0	Out	X PWM0 drive command
31A	X_EN	Out	X drive Enable
32A	5VF	Sup	Field 5V
1B)	X_A	Out	X Analog output. ±10V
2B	Y_A	Out	Y Analog output. ±10V
3B	Z_A	Out	Z Analog output. ±10V
4B	T_A	Out	T Analog output. ±10V
5B	A_OUT	Out	General purpose analog output. ±10V
6B	AI1	In	Analog in #1 (Joystick). 0 - 5V
7B	AI2	In	Analog in #2 (Joystick). 0 - 5V
8B	N.C.		
9B	N.C.		
10B	N.C.		
11B	N.C.		
12B	N.C.		
13B	N.C.		
14B	ES	In	Emergency Stop
15B	X_LL	In	X Left Limit
16B	X_RL	In	X Right Limit
17B	OUT8	Out	Out #8
18B	OUT7	Out	Out #7
19B	OUT6	Out	Out #6
20B(*)	OUT5	Out	Out #5
21B	OUT4	Out	Out #4
22B	OUT3	Out	Out #3
23B	OUT2	Out	Out #2
24B	OUT1	Out	Out #1
25B	Z_STL	Out	Z Settling. (For four phase steppers only)
26B	Z_FLT	In	Z drive Fault feedback
27B	Z_PWM0	Out	Z PWM0 drive command
28B	Z_PWM1	Out	Z PWM1 drive command

Pin	Signal	I/O	Description
29B	Z_EN	Out	Z drive Enable
30B	X_PWM1	Out	X PWM1 drive command
31B	+12VF	Sup	Field +12V
32B	RTNF	Sup	Field Return
1C	T_CH2I_DS	In	T Encoder #2 disconnected
2C	T_CH2B	In	T Encoder #2 B
3C	T_CH2A	In	T Encoder #2 A
4C	T_CH2I	In	T Encoder #2 Index
5C	Z_CH2I_DS	In	Z Encoder #2 disconnected
6C	Z_CH2B	In	Z Encoder #2 B
7C	Z_CH2A	In	Z Encoder #2 A
8C	Z_CH2I	In	Z Encoder #2 Index
9C	YCH2I_DS	In	Y Interferometer disconnected
10C	YSCLK	Out	Y clock
11C	YCONVST	Out	Y Start
12C	YADC_DI	In	Y Data In
13C	XCH2I_DS	In	X Interferometer disconnected
14C	XSCLK	Out	X clock
15C	XCONVST	Out	X Start
16C	XADC_DI	In	X Data In
17C	IN8	In	Input #8
18C	IN7	In	Input #7
19C	IN6	In	Input #6
20C	IN5	In	Input #5
21C	IN4	In	Input #4
22C	IN3	In	Input #3
23C	IN2	In	Input #2
24C	IN1	In	Input #1
25C	T_STL	Out	T Settling command. (For four phase steppers only)
26C	T_FLT	In	T drive Fault feedback
27C	T_PWM0	Out	T PWM0 drive command
28C	T_PWM1	Out	T PWM1 drive command
29C	T_EN	Out	T drive Enable
30C	X_FLT	In	X drive Fault feedback
31C	X_STL	Out	X Settling command. (For four phase steppers only)
32C	-12VF	Sup	Field -12V

5.3.14 J4 (P1) Connector

VME bus interface.

Pin	Signal	I/O	Description
1A	D0	I/O	Data 0
2A	D1	I/O	Data 1
3A	D2	I/O	Data 2
4A	D3	I/O	Data 3
5A	D4	I/O	Data 4
6A	D5	I/O	Data 5
7A	D6	I/O	Data 6
8A	D7	I/O	Data 7
9A	GND	Sup	Ground
10A	N.C.		
11A	GND	Sup	Ground
12A	DS1N	In	Data Strobe 1
13A	DS0N	In	Data Strobe 0
14A	WRITEN	In	Write
15A	GND	Sup	Ground
16A	DTACKN	Out	Data Transfer Acknoledge
17A	GND	Sup	Ground
18A	ASN	In	Address Strobe
19A	GND	Sup	Ground
20A	IACKN	In	Interrupt Acknowledge
21A	IACKINN	In	Interrupt Acknowledge In
22A	IACKOUT	Out	Interrupt Acknowledge Out
23A	AM4	In	Address Modifier 4
24A	A7	In	Address 7
25A	A6	In	Address 6
26A	A5	In	Address 5
27A	A4	In	Address 4
28A	A3	In	Address 3
29A	A2	In	Address 2
30A	A1	In	Address 1
31A	-12V	Sup	-12V Power
32A	+5V	Sup	+5V Power
1B	N.C.		
2B	N.C.		
3B	N.C.		
4B	N.C.		
5B	N.C.		
6B	N.C.		
7B	N.C.		
8B	N.C.		

Pin	Signal	I/O	Description
9B	N.C.		
10B	N.C.		
11B	N.C.		
12B	N.C.		
13B	N.C.		
14B	N.C.		
15B	N.C.		
16B	AM0	In	Address Modifier 0
17B	AM1	In	Address Modifier 1
18B	AM2	In	Address Modifier 2
19B	AM3	In	Address Modifier 3
20B	GND	Sup	Ground
21B	N.C.		
22B	N.C.		
23B	GND	Sup	Ground
24B	IRQN7	Out	Interrupt request 7
25B	IRQN6	Out	Interrupt request 6
26B	IRQN5	Out	Interrupt request 5
27B	IRQN4	Out	Interrupt request 4
28B	IRQN3	Out	Interrupt request 3
29B	IRQN2	Out	Interrupt request 2
30B	IRQN1	Out	Interrupt request 1
31B	N.C.		
32B	+5V	Sup	+5V Power
1C	N.C.		
2C	N.C.		
3C	N.C.		
4C	N.C.		
5C	N.C.		
6C	N.C.		
7C	N.C.		
8C	N.C.		
9C	GND	Sup	Ground
10C	N.C.		
11C	N.C.		
12C	SYSRESTN	In	System Reset
13C	LWORDN	In	Long word
14C	N.C.		
15C	N.C.		
16C	N.C.		
17C	N.C.		
18C	N.C.		

Pin	Signal	I/O	Description
19C	N.C.		
20C	N.C.		
21C	N.C.		
22C	N.C.		
23C	A15	In	Address 15
24C	A14	In	Address 14
25C	A13	In	Address 13
26C	A12	In	Address 12
27C	A11	In	Address 11
28C	A10	In	Address 10
29C	A9	In	Address 9
30C	A8	In	Address 8
31C	+12V	Sup	+12V Power
32C	+5V	Sup	+5V Power

5.3.15 SB214VME Jumpers

See Figure 5-25.

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Default R/IN	Remarks
JP1	Read by Master µP HSI.3			R	For future use 2 pins jumper.
JP2	Read by Master µP HSI.2			R	For future use 2 pins jumper
JP3	Read by Master µP HSI.1			R	For future use 2 pins jumper
JP4	Slave µP ROM	external ROM	internal ROM	In	2 pins jumper, For factory use only
JP5	Clock source	system clock	separate clock	In	For factory use only 2 pins jumper
JP6	PPI interrupt source	Master µP HSO.0	from SB2500	R	For factory use only 2 pins jumper
JP7	HW reset	Enable reset via RS232 break	Disable reset	R	For factory use only 2 pins jumper
JP8	HW watchdog	Disabled	Enabled	2-3	For factory use only
JP9	Master µP reset	Delayed reset	System's reset	1-2	For factory use only
JP10	Extint source	Master µP HSO.0	Slave µP HSO.1	1-2	For factory use only
JP11	SB2500 interrupt source	Slave µP HSO.0	SB2500	R	For factory use only 2 pins jumper
JP12	Master µP bus access	Free	Using arbitration	1-2	For factory use only
JP13	Slave µP bus access	Using arbitration	No access	2-3	For factory use only
JP14	Non-volatile memory power source	Installed - From VME	Removed - local supply	R	For factory use only 2 pins jumper
JP15	X Command 0	PWM/Pulse	Analog	2-3	
JP16	Y Command 0	PWM/Pulse	Analog	2-3	
JP17	Z Command 0	PWM/Pulse	Analog	2-3	
JP18	T Command 0	PWM/Pulse	Analog	2-3	
JP19	Output common	VME ground. No isolation	External RTNF. Isolation	2-3	

Jumper	Description	'UP' (1-2) (Installed)	'DOWN' (2-3) (Removed)	Default R/IN	Remarks
JP20	Output supply	VME 5V	Field 5VF	2-3	
JP21	Input, limits, ES pull-up	VME 5V	Field 5VF	2-3	

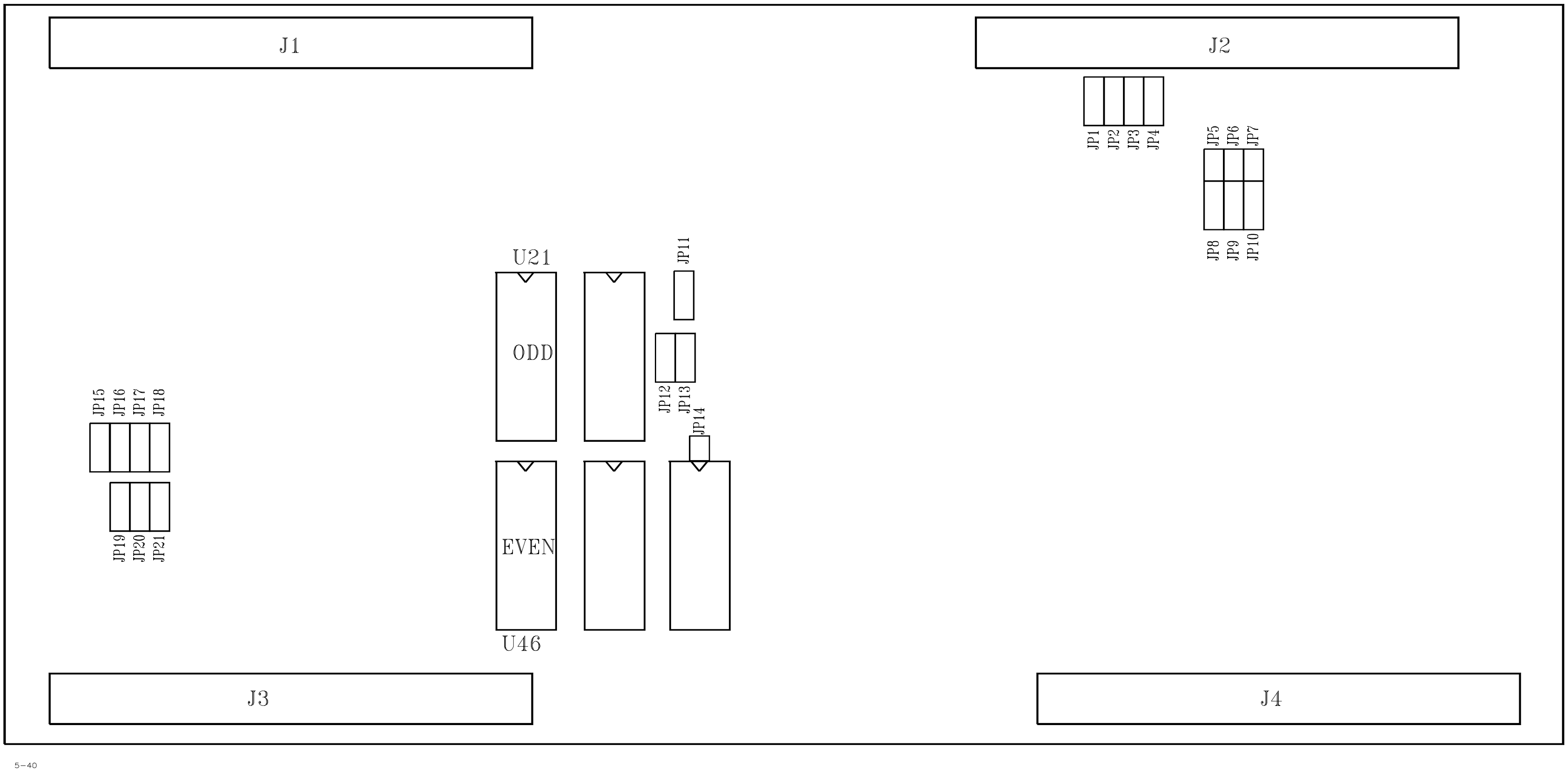

Figure 5-25 SB214VME Jumpers

5.3.16 User's Options Set Up

External Hardware Reset

This feature enables the user to reset the board externally, using the RXD receive line. It can be done by either one of the following ways:

- Forcing a break command on the RS232 of the Host computer.
- Short the RS232 RXD to +12V using an external switch.

This option is enabled by installing JP7.

Driver Interface

Commands 0 of each axis can be set to a ±10V analog command, or 0-5V PWM type command. The PWM command is used for drivers with PWM input or Pulse and Direction input.

JP15-JP18 select the type of command for a specific axis.

Isolation And External Supplies

The drivers, the input port and the output port can be isolated from the controller/VME by setting the appropriate jumpers and using external isolated supplies. The following matrix describes the relation between the setting of the jumpers and the various isolations. Note that if there is no isolation between two isolated parties. For example, if the input port and the drive control are isolated, then there is no isolation between the drive and the input.

	Drivers	**Input port**	**Output port**	**JP19**	**JP20**	**JP21**
1	IS	IS	IS	2-3	2-3	2-3
2	IS	IS	NI	1-2	1-2	2-3
3	IS	NI	IS	2-3	2-3	1-2
4	IS	NI	NI	1-2	1-2	1-2
5(*)	NI	NI	NI	1-2	1-2	1-2

IS - Isolated

NI - No Isolation

(*) In case 5, the following supply lines must be connected by the user:

5V⇔5VF

+12VF ⇔ +12V

-12VF ⇔ -12V

RTNF ⇔ GND

5.3.17 VME Bus Interface

The board is a slave type. It uses "Single odd byte (8 bits) data transfer" mode.

Address Modifier (AM5 - AM0):

- Non-Privileged Access (29H), or
- Short Supervisory Access (2DH).

The VME bus has 32 address lines A31-A0. Lines A31-A16 are defined by the VME system. Those lines do not reach the controller. Within the 64K address space that is defined by the VME system, the card offset is defined by A15-A8 via the on board DIP switch (A15-SW1, A8-SW8). **A15 (SW1) must be set to "on" (when SW1 is "off" the card communicates via the serial link)**. The SB214VME are factory set to communicate over the VME address 00H. Within the 256 bytes address space that is defined by A15-A8, the address of the four communication registers are:

61H TX_FIFO, Read from controller's FIFO.

41H RX_FIFO, Write to controller's FIFO.

21H FLAGS, Read controller's flags register.

01H STATUS, Read controller's status register.

FLAGS - Flags register bits:

D0 TX_FIFO Full.

D TX_FIFO Empty.

D2 RX_FIFO Full.

D3 RX_FIFO Empty.

The flags are active low.

5.4 SB214ND Stand Alone 4 Axis Controller

Warning: Do not open the box while power is applied. Turn the power off and wait for at least one minutes until the internal capacitors are fully discharged.

5.4.1 Installation

In order to meet the safety requirements of EN60204-1 the module must be installed in a cabinet or a similar enclosure with protection level IP4X. (Figure 5-26)

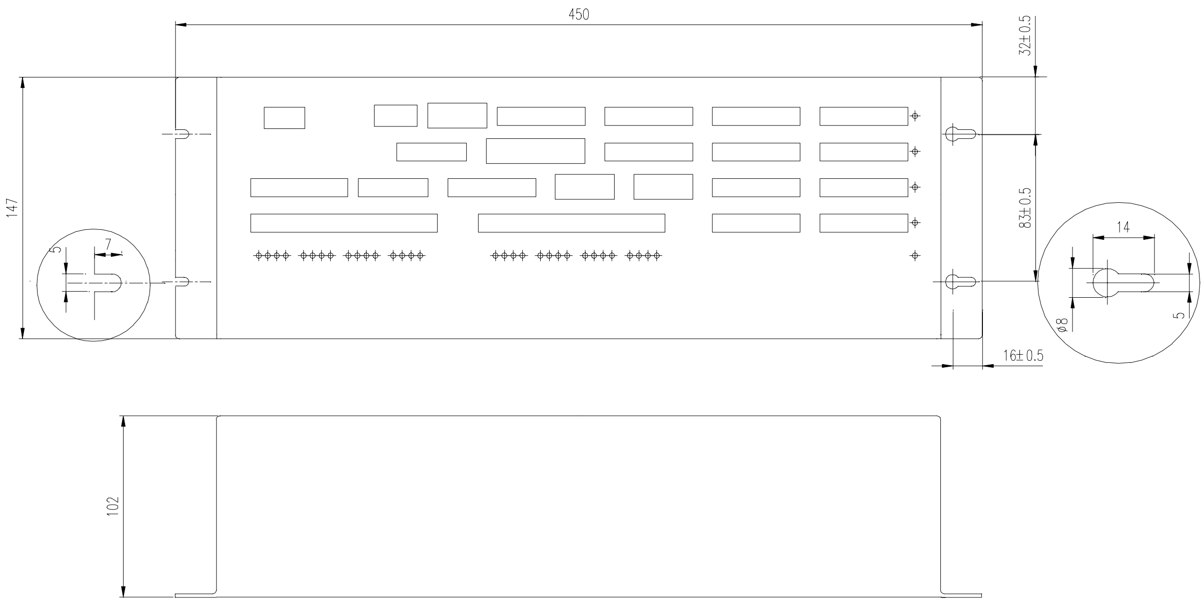

Figure 5-26 SB214ND dimensions and mounting

The ambient temperature must not exceed 45ºC. If necessary, forced air exchange must be allowed for. The thermal design of the cabinet should take into account that at the nominal power, the unit dissipates up to 20W.

- Do not mount modules on top of each other.
- Allow for at least 15mm clearance bellow and above the unit. (There is no need for side clearance.)
- The environment must be kept free from corrosive chemical vapors, oil steam, metal particles and dust.

5.4.2 Preparation Of Cables For Phoenix Connectors

Do not solder the tips of the cables before insertion into the connector. Solder will contract and cause a loose connection over time.

All wires must be stripped to 0.27" (7mm). (Figure 5-27)

24 to 12 gauge wires can be used, unless otherwise stated.

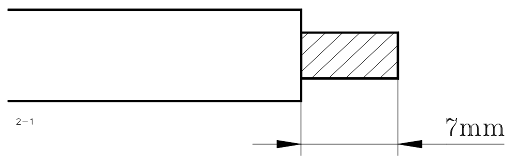

Figure 5-27 Wire stripping dimension

5.4.3 Power Supplies

Three internal power supplies are supporting the various functions of the system. The Controller Supply (5VPC, ±12V), the Field Isolation Supply (5VF±12VF), and the Output Port Supply (24VF1, 24VF2).

Controller Power Supply 5VPC, +12V,-12V	It feeds all the internal circuits of the system. Its voltages are related to **COM**.
Field Power Supply 5VF, 1.0A +12VF,0.15A -12VF, 0.15A	It supplies the power to the isolated (field) side of the interface to the drivers. Its voltages are related to **RTNF.**
Output Port Supply 24VF1, 24VF2 total 1.0A	It supplies the power to the output port. It may also be used to support the input port. The 24V is related to **RTNF.** OUT1-8 are powered by the 24VF1 input pin. OUT9-16 are powered by the 24VF2 pin. Internally, both 24VF1 and 24VF2 are connected to the 24V supply. Fuse: 250V/1.6A. (Located on the interface card.)

5.4.4 S1 DIP Switch

The S1 dip switch is located at the top of the module.

SW1 must be in "off" position

SW4 is used for partial communication shutdown. See page 6-2.

Off	Full communication.
On	Partial shut-down. The communication link is exclusively under the ACSPL program control. Only ‘disp’ message are transmitted, and only responses to ‘input’ statements are processed.

All other switches are for future use

5.4.5 Led Indicators

mP ON	On power up (reset), this LED flickers for a while. Afterwards, whenever the controller detects an RS232C receive message, it turns the LED off for a fraction of a second, and afterwards turns on again. It is used as an indicator that the processor and communication function properly.
MOTOR ON	These LEDs signal that the motors are powered. Whenever the drives are disabled, either as a result of user disable command, or as a result of an erroneous situation (over current, over temperature, large error etc.), the LED is turned off. There are four LEDs, one for each axis: X,Y,Z and T.
Input port LEDs	When current is flowing through the input port opto-coupler, the appropriate LED is turned on. (Note that ‘off’ and ‘on’ are defined by the INput logic parameter and may not correspond to the LED being ‘off’ and ‘on’.)
Output port LEDs	When an output is turned on the appropriate LED will light.

KEYING: Most of the connectors are keyed in order to prevent mismatching of connectors. Each unit is shipped with the keys installed. The warranty does not cover any damage that is caused by mismatching.

5.4.6 AC Power Connector (110Vac / 230Vac)

Connector: Phoenix DFK-MSTB 2.5/4-G.
Mating - MSTB 2.5/4-ST

Pin	Name	Description
4	NC	Not Connected
3	GND	Chassis ground
2	AC+	110/230VAC hot, fused. -20% +10%
1	AC-	110/230VAC neutral

.

The AC input supply can be single phase, 110Vac (-15%+10), or 230Vac(-15%+10%).

The unit is set at the factory for either 110Vac input or for 230Vac. The setting is indicated on the label.

Input current

110Vac	< 200mA
230Vac	< 100mA

It is recommended to use 16 gauge wire for the power supply cable.

The AC+ hot input (#2) is fused to avoid fire hazard. The AC hot line must be connected to input #2 (AC+)!
Units are factory adjusted for either 110, or 230 VAC supply. The line voltage is marked on the label on the left side panel of the module.

110VAC Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
NC	4	-	-	4
GND	3	-	X	3
AC+	2	X	-	2
AC-	1	-	X	1

230VAC Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
NC	4	-	-	4
GND	3	X	-	3
AC+	2	-	X	2
AC-	1	-	X	1

5.4.7 External Supply Connector

Connector type: Phoenix DFK-MSTB 2.5/7-G.
Mating: MSTB 2.5/7-ST

Pin #	Name	Description
1	5VF	Field 5V
2	+12VF	Field +12V
3	-12VF	Field -12V
4	RTNF	Field return (common)
5	5VPC	5V of the PC. Do not use !
6	5V	This 5V powers the interface.
7	COM	5V, 5VPC return (common)

External Supply Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5VF	1	X	-	1
+12VF	2	-	X	2
-12VF	3	-	-	3
RTNF	4	-	-	4
5VPC	5	X	-	5
5V	6	-	X	6
COM	7	-	X	7

5.4.8 Input Port

Connector type: Phoenix MSTB 2.5/20-G1.
Mating - MSTB 2.5/20-ST.

It is recommended to add keys to the I/O connectors, in order to prevent mismatching. The user must define the locations of the keys.

There are 16 general purpose inputs IN1 - IN16. All inputs are opto-coupled. 10MHz opto-couplers are used for the registration inputs (6,7,9,10).

The fast registration inputs' filter (RC) has a time constant of 1 μsec. The rest of the inputs have a time constant of approximately 1 msec.

The state of the input is interpreted by the controller based on the input level and on the corresponding IN (INput logic) bit. When the IN bit is 0, then the input is 'on' when the input pin is shorted to RTNF.
The registration inputs activate the AUTO_M1 routines and cause latching of the mark position when shorting the input to RTNF independent of the IN bit setting.

To power the port you can use one of the built-in supplies, 5VF or 24VF, or an external supply. When using an internal supply, connect the 5VF or the 24VF to VIN. When using an external supply, connect it to the VIN pin. To activate and deactivate an input, short the input to the return of the external supply.

IN1 to IN5 are supported by ACSPL automatic routines (AUTO_IN1 - AUTO_IN5).

The **ACSPL IP** parameter holds the binary representation of the 16 inputs status.

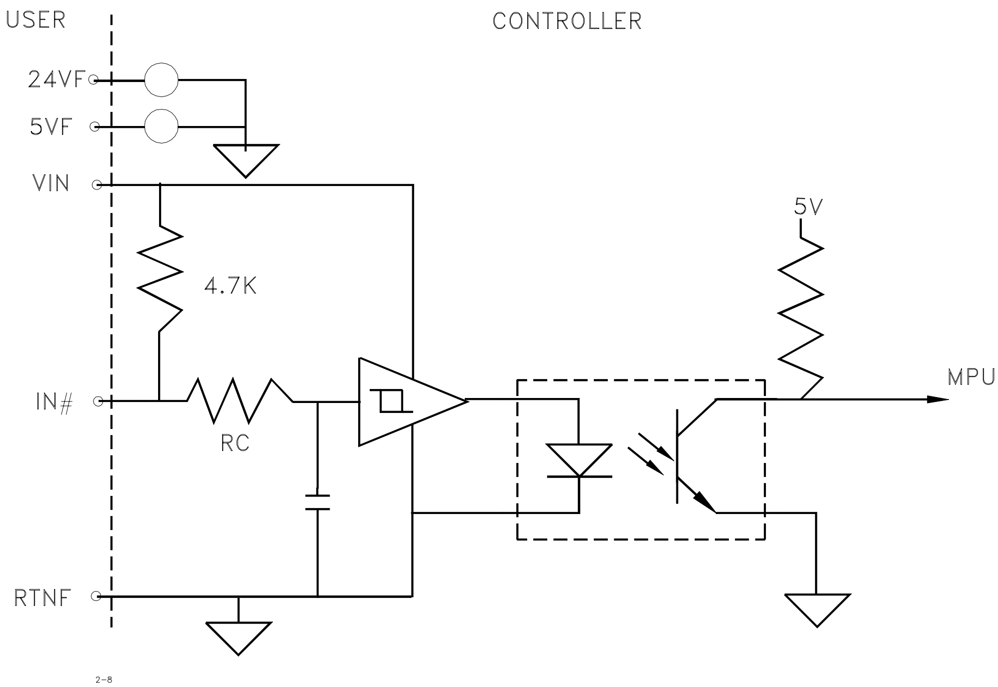

Figure 5-28 Input port

The fast registration inputs (6,7,9,10) filter (RC) has a time constant of 1 µ sec. The rest of the inputs have a time constant of 5 µsec.

Pin #	Name	Description
1	RTNF	Input RETURN
2	IN1	Input 1
3	IN2	Input 2
4	IN3	Input 3
5	IN4	Input 4
6	VIN	All inputs are pull-up to VIN
7	IN5	Input 5
8	IN6	Input 6 X registration mark
9	IN7	Input 7 Y registration mark
10	IN8	Input 8
11	5VF	Field 5V can be used as VIN
12	IN9	Input 9 Z registration mark
13	IN10	Input 10 T registration mark
14	IN11	Input 11
15	IN12	Input 12
16	24VF1	Field 24V can be used as VIN
17	IN13	Input 13
18	IN14	Input 14
19	IN15	Input 15
20	IN16	Input 16

5.4.9 Output Port

Connector type: Phoenix MSTB 2.5/20-G1.
Mating: MSTB 2.5/20-ST.

There are 16 general purpose digital outputs available to the user. Outputs may be programmed to serve a dedicated function like signaling of motion execution. See Figure 5-29.

The outputs are opto isolated. They are built around ULN2803A device which consists of 8 Darlingtons with free wheeling diodes that are connected to the 24VF pin.
Outputs 1-8 are built around one ULN2803A with the diodes connected to 24VF1, and outputs 9-16 around the other with the diodes connected to 24VF2.
Each output is capable of sinking up to 0.1A. Total sinking must be limited to 0.5A per ULN2803A device.
The internal 24VF, 1A supply is available to the user for small inductive loads driving. Both 24VF1 and 24VF2 are connected to the internal 24VF supply.

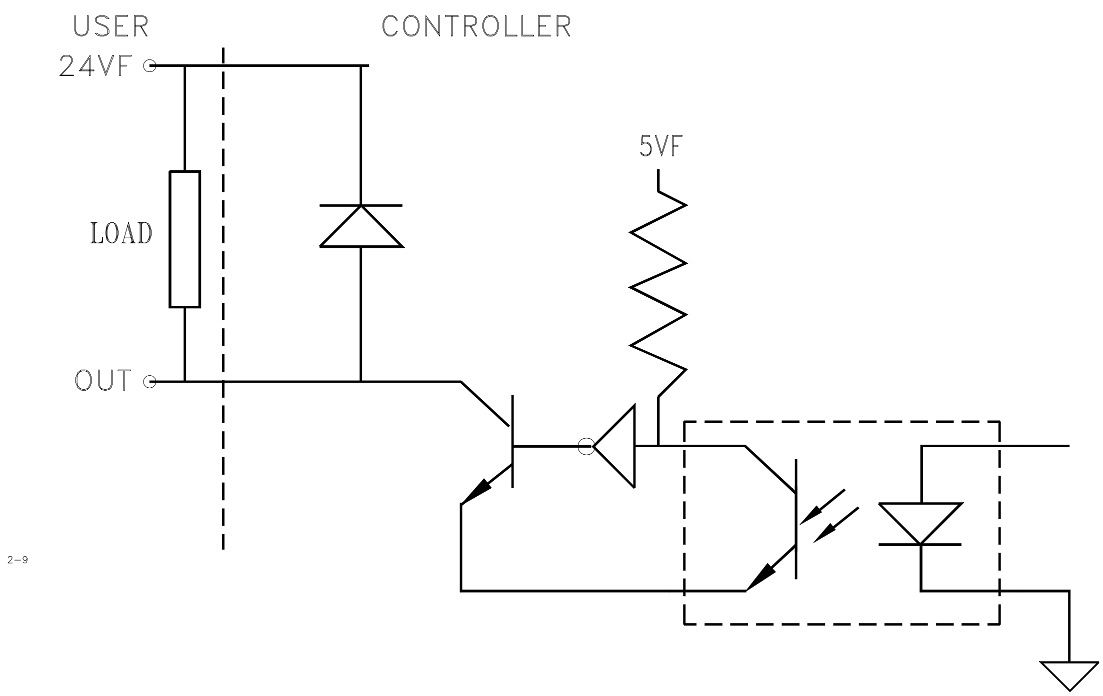

Figure 5-29 Output port interface

Pin #	Name	Description
1	OUT1	Output 1
2	OUT2	Output 2
3	OUT3	Output 3
4	OUT4	Output 4
5	RTNF	Output RETURN
6	OUT5	Output 5
7	OUT6	Output 6
8	OUT7	Output 7
9	OUT8	Output 8
10	24VF1	24V supply for output ports 1 - 8
11	OUT9	Output 9
12	OUT10	Output 10
13	OUT11	Output 11
14	OUT12	Output 12
15	RTNF	Output RETURN
16	OUT13	Output 13
17	OUT14	Output 14
18	OUT15	Output 15
19	OUT16	Output 16
20	24VF2	24V supply for output ports 9 - 16

5.4.10 Safety Inputs

Connector type: Phoenix MSTB 2.5/10-G1.
Mating: MSTB 2.5/10-ST.

The EMERGENCY STOP input is used to stop and deactivate the axis instantaneously.
The limit inputs are used to avoid axis over travel by preventing further motion in the inhibited direction.
The structure of the safety inputs is identical to the structure of the general purpose inputs.
The polarity of the safety inputs can be altered by the Input Logic (IL) parameter.

Pin #	Name	Description
1	ES	Emergency Stop
2	RTNF	Return
3	LLT	T Left Limit input
4	RLT	T Right Limit input
5	LLZ	Z Left Limit input
6	RLZ	Z Right Limit input
7	LLY	Y Left Limit input
8	RLY	Y Right Limit input

9	LLX	X Left Limit input
10	RLX	X Right Limit input

Safety Inputs Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
ES	1	X	-	1
RTNF	2	-	X	2
LLT	3	-	-	3
RLT	4	-	-	4
LLZ	5	-	-	5
RLZ	6	-	-	6
LLY	7	-	-	7
RLY	8			8
LLX	9	-	X	9
RLX	10	X	-	10

5.4.11 Analog / Joystick Inputs

Connector type: Phoenix MSTB 2.5/9-G1.
Mating: MSTB 2.5/9-ST

There are 6 Analog Inputs - AI1 - AI6. Analog Inputs 1 (ACSPL's A0) and 2 (ACSPL's A1) can be used as joystick inputs. AI1 for X and Z, and AI2 for Y and T.

The joystick's potentiometer must be connected between the '5R' pin and the COM pin. The potentiometer output is connected to AI1 and AI2 pins.
The two inputs act also as general purpose analog inputs that are sampled via a 10 bits ADC by the A0 and A1 ACSPL variables. 0V is represented by 0, and 5V is represented by 1024. The user must make sure that the input voltage (relative to the COM pin) does not exceed the specified range of [0 - 5V].

Inputs AI3 to AI6 are sampled by a 12 bits ADC.

The voltage range is +/-10V. +10V is represented by 2048. -10V is represented by -2048.

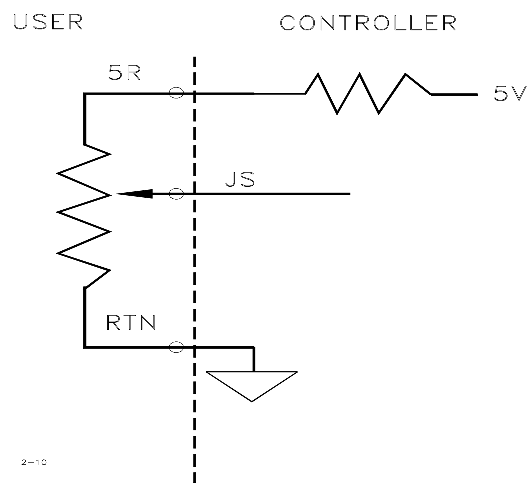

Figure 5-30 Joystick connection

Pin #	Name	Description
1	5R	5V Supply to Joystick
2	AI6	Analog Input 6
3	AI5	Analog Input 5
4	AI4	Analog Input 4
5	AI3	Analog Input 3
6	AI2	Analog Input 2
7	AI1	Analog Input 1
8	COM	Common Ground
9	SCRN	Shield

Analog Inputs Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5R	1	-	X	1
AI6	2	-	X	2
AI5	3	X	-	3
AI4	4	-	-	4
AI3	5	-	-	5
AI2	6	X	-	6
AI1	7	-	X	7
COM	8	-	X	8
SCRN	9	-	-	9

5.4.12 Analog Outputs

There are five Analog Outputs, in the range -10V ÷ +10V.
The analog output (**A_OUT**) is an 8 bit general purpose that can be set via the **D0** parameter.
There is one 10 bits analog output for each axis, to be used for monitoring

purposes, or as an uncommitted, general purpose analog output - **X_A (XD4), Y_A(YD4), Z_A(ZD4), T_A(TD4).**

To use an axis output as an uncommitted output, set $MN to 0. For example, to use XD4 - **SXMN0<cr>.**

To use an axis output to monitor a real time signal, set $MN to 1 (velocity), 2 (current command), or 3. When $MN=3, $DC points to the required parameter. For example, to monitor the position error of Y axis - **SYMN3 DC4<cr>.**

Pin #	Name	Description
1	COM	Common (ground)
2	T_A	T Analog Out - TD4
3	Z_A	Z Analog Out - ZD4
4	Y_A	Y Analog Out - YD4
5	X_A	X Analog Out - XD4
6	A_O	General purpose Analog Output - D0
7	SCRN	Shield

Analog Outputs Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
COM	1	X	-	1
T_A	2	-	X	2
Z_A	3	-	-	3
Y_A	4	-	-	4
X_A	5	-	X	5
A_O	6	-	X	6
SCRN	7	X	-	7

5.4.13 Encoder Connectors

Connector type: Phoenix MSTB 2.5/9-G1.
Mating: MSTB 2.5/9-ST.

There are six encoder ports - encoder #1 (primary) for X,Y,Z and T, and encoder #2 (secondary) for Z and T. Each one can serve a different function according to the setup of the specific axis.

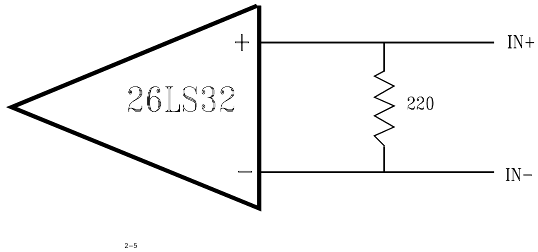

Figure 5-31 Encoder interface

Each encoder input port accepts three channel differential, TTL level, optical encoder, or laser interferometer signals of various formats. The format and the maximum counting speed is programmable via the Encoder Type (**ET** - primary, **et** - secondary) parameters.

- UP-DOWN plus index.
  The maximum counting rate is either 10 MHz ($ET=0), or 20 MHz ($ET=100).
- CLOCK-DIR plus index.
  The maximum counting rate is either 10 MHz ($ET=1), or 20 MHz ($ET=101).
- Two adjacent bits (C0, C1) of an up-down counter plus index.
  The maximum counting rate is either 10 MHz ($ET=2), or 20 MHz ($ET=102).
- Two phase quadrature plus index. (Factory default, 10MHz).
  The maximum counting rate (including X4) is either 10 MHz ($ET=3), or 20 MHz ($ET=103).

The input buffer is built around AM26LS32 line receivers. It is recommended to use encoders with built-in line drivers (AM26LS31 or similar).

Pin #	Name	Description
1	5V	5V, fused by 1.6A fuse
2	A+	Encoder A input (CLOCK, UP, C0)
3	A-	Encoder inverted A input (CLOCK, UP, C0)
4	B+	Encoder B input (DIR, DOWN, C1)
5	B-	Encoder inverted B input (DIR, DOWN, C1)
6	I+	Encoder Index input
7	I-	Encoder inverted Index input
8	COM	Common (Ground)
9	SCRN	Shield

X Encoder#1 Keying

(**X** - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	**X**	-	1
A+	2	-	X	2
A-	3	-	X	3
B+	4	-	-	4
B-	5	-	X	5
I+	6	-	X	6
I-	7	X	-	7
COM	8	-	X	8
SCRN	9	-	-	9

Y Encoder#1 Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	X	-	1
A+	2	-	X	2
A-	3	-	X	3
B+	4	-	-	4
B-	5	-	X	5
I+	6	-	X	6
I-	7	-	X	7
COM	8	X	-	8
SCRN	9	-	-	9

Z Encoder#1 Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	X	-	1
A+	2	-	X	2
A-	3	-	X	3
B+	4	-	-	4
B-	5	-	X	5
I+	6	X	-	6
I-	7	-	X	7
COM	8	-	X	8
SCRN	9	-	-	9

T Encoder#1 Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	X	-	1
A+	2	-	X	2
A-	3	-	X	3
B+	4	-	-	4
B-	5	X	-	5
I+	6	-	X	6
I-	7	-	X	7
COM	8	-	X	8
SCRN	9	-	-	9

Z Encoder#2 Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	-	X	1
A+	2	-	X	2
A-	3	X	-	3
B+	4	-	-	4
B-	5	-	-	5
I+	6	-	X	6
I-	7	-	X	7
COM	8	X	-	8
SCRN	9	-	-	9

T Encoder#2 Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
5V	1	-	X	1
A+	2	-	X	2
A-	3	X	-	3
B+	4	-	-	4
B-	5	-	-	5
I+	6	-	X	6
I-	7	X	-	7
COM	8	-	X	8
SCRN	9	-	-	9

5.4.14 High Speed Serial Interface ("Interferometer")

Connector type: D-Type Female 9 pins.
Mating: D-Type Male 9 pins.

There are two, general purpose high speed synchronous serial interface ports. They can be used for high speed laser interferometers position reading, or any other function as required by the user. Consult factory if you need to use them. They are labeled X and Y ports #2. The interface is implemented using 3 differential signals: Start, Clock and Data In. See Figure 5-32.

Pin #	Name	Description
1	COM	Common Ground
2	START+	Start (Output)
3	START-	Inverted Start (Output)
4	5V	5V, fused by 1.6A fuse
5	SCRN	Shield
6	CLOCK+	Clock (Output)
7	CLOCK-	Inverted Clock (Output)
8	DATA IN+	Data In (Input)

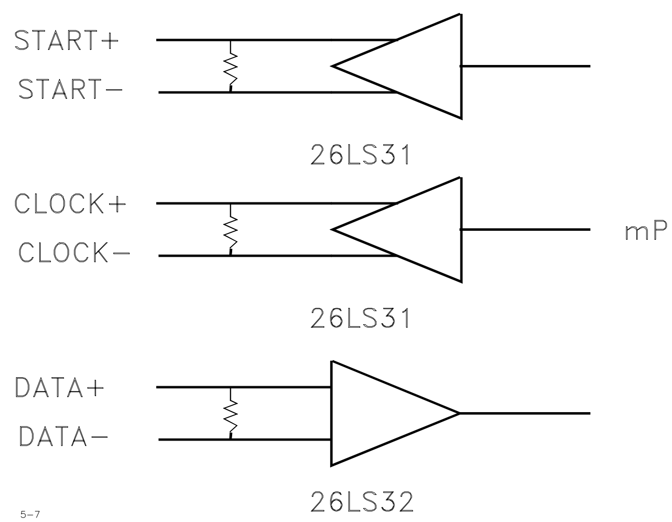

Figure 5-32 High Speed Serial Interface

5.4.15 Serial Communication Interface

Connector type: Phoenix MSTBA 2.5/7-G.

Mating: MSTB 2.5/7-ST.

Two hardware interfaces are supported:

RS232.

RS422.

No Jumpers or Switches are needed for setup. Just connect the signals of the type of interface that you intend to use.

Pin #	Name	Description
1	COM	Common.
2	R232	RS232 Receive (input).
3	T232	RS232 Transmit (output).
4	-R422	RS422 Inverted Receive (input).
5	+R422	RS422 Receive (input).
6	-T422	RS422 Inverted Transmit (output).
7	+T422	RS422 Transmit (output).
8	5V	5V supply to a hand held terminal.
9	SCRN	Shield.

Serial Communication Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
COM	1	X	-	1
R232	2	-	X	2
T232	3	-	-	3
-R422	4	-	-	4
+R422	5	-	-	5
-T422	6	-	-	6
+T422	7	-	X	7
5V	8	X	-	8
SCRN	9	-	X	9

Linking More Than One Module

Up to 10 units of SB214ND and SB1000 control modules can be connected to one RS422 serial link.

Follow this procedure:

- Assign a different ID number to each module:
- Connect the first unit to a computer.
- Set its ID to 0 (**SID0<cr>)**.
- Execute a SAVE command (**SAVE<cr>** twice.)

The new ID becomes effective after the next power up.

- Connect the second unit.
- Set its ID to 1 (**SID1<cr>).**
- Execute a SAVE command (**SAVE<cr>** twice.).

The new ID becomes effective after the next power up.

Repeat the above for each additional unit up to ID=9.

- Connect the 'Transmit (TX)' of all modules in parallel.

- Connect the 'Receive (RX)' of all modules in parallel.
- Connect the 'COM' of all modules in parallel. (Figure 5-33)

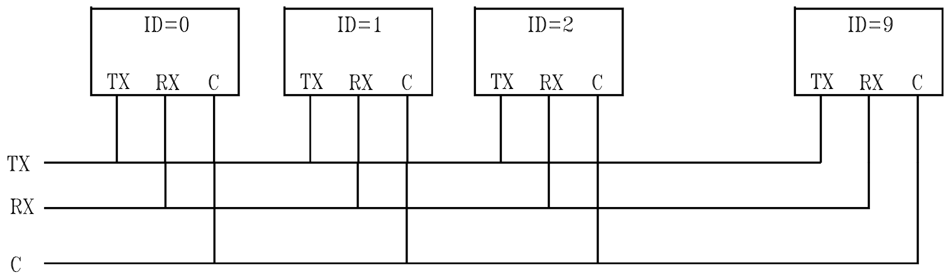

Figure 5-33 Linking more than one SB214 module to one serial link

5.4.16 Drive Interface

Connector: Phoenix DFK-MSTB 2.5/9-G.
Mating: MSTB 2.5/9-ST.

The controller can control the following types of motors/drivers:

- A stepper motor driver with two phase current commands (XT=0).
- A servo motor with single torque/velocity command (XT=1).
- A stepper motor driver with Pulse-Dir command (XT=2).
- A sinusoidal brushless motor with a drive that accepts two phase current commands (the controller does the commutation) (XT=3).
- An AC induction motor with a drive that accepts two phase current commands, using Field Oriented Control (Vector Control) method (XT=5).

Each axis supprts one and two output commands, in two formats: PWM (or Pulse-Dir), and analog (V#).

When the drive-motor system belongs to group 1 (see page 4-1), PWM0 and V0 are used.

When the drive-motor system belongs to group 2 (see page 4-1), PWM0, PWM1 or V0, V1 are used.

When the drive-motor system belongs to group 3 (see page 4-1), Pulse -Dir are used.

All signals are opto-isolated from the digital controller.

For drives that accept single ±10V command, use V0.

For Drives that accept single PWM type command, use PWM0.

For Standard step motor drives, use PWM0 (Pulse) , PWM1(Dir).

For two, three and four phase brushless and AC drives that accept two phase analog current commands, use V0,V1

For two, three and four phase brushless and AC drives that accept two phase PWM type current commands, use PWM0, PWM1.

For ACS micro stepper drives (ST-4), use PWM0, PWM1.

Pin #	Name	Description
1	PWM1	Command 1,PWM format. 0-5V, 50% duty cycle when the command DO is zero. 99.9% when DO=511, 0.1% when DO=0. Dir command for stepper.
2	PWM0	Command 0 in PWM format. 0-5V.50% duty cycle when the command DO is zero. 99.9% when DO=511, 0.1% when DO=0. Pulse (clock) command for stepper. The pulse width is programmable by $PW parameter. Range - [0.4 - 60] microsecond. Rate - [1 - 2,000,000] Hz.
3	STL	Settling. Active low. Used with ACS 4 phase stepper drivers only.
4	FLT	Fault feedback from drive. Upon fault, this input should be connected to the common pin via a low resistance switch. The logic can be changed via IL parameter. When FLT becomes TRUE E26 message is generated.
5	EN-	Inverted enable. The emitter of the opto-transistor. When enabled, the transistor is on.
6	EN+	Enable. The collector of an opto-transistor. The logic can be changed via OL parameter.
7	RTNF	This is the common pin to all the signals between the drive and the controller. It should be connected to the common of the drive.
8	V1	Command 1.Analog format, ±10V range.
9	V0	Command 0. Analog format ±10V range.

X Drive Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
PWM1	1	-	X	1
PWM0	2	X	-	2
STL	3	-	X	3
FLT	4	-	-	4
EN-	5	-	X	5
EN+	6	-	X	6
RTNF	7	X	-	7
V1	8	-	X	8
V0	9	-	-	9

Y Drive Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
PWM1	1	-	X	1
PWM0	2	X	-	2
STL	3	-	X	3
FLT	4	-	-	4
EN-	5	-	X	5
EN+	6	-	X	6
RTNF	7	-	X	7
V1	8	X	-	8
V0	9	-	-	9

Z Drive Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
PWM1	1	-	X	1
PWM0	2	X	-	2
STL	3	-	X	3
FLT	4	-	-	4
EN-	5	-	X	5
EN+	6	X	-	6
RTNF	7	-	X	7
V1	8	-	X	8
V0	9	-	-	9

T Drive Keying

(X - KEY INSTALLED)

Name	Pin	On module	Mating	Pin
PWM1	1	-	X	1
PWM0	2	X	-	2
STL	3	-	X	3
FLT	4	-	-	4
EN-	5	X	-	5
EN+	6	-	X	6
RTNF	7	-	X	7
V1	8	-	X	8
V0	9	-	-	9

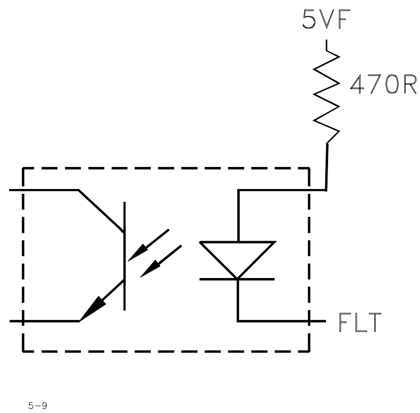

Figure 5-34 Drive fault input

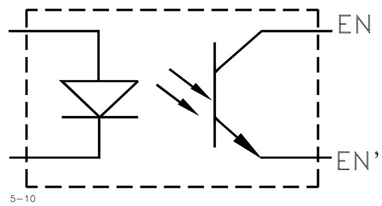

Figure 5-35 Drive enable output

5.4.17 ACS Drive Box

Connector type: D-Type Male 25 pins.
Mating: D-Type Female 25 pins.

This interface is specially designed for interfacing with ACS made multi axis drive modules, like the DR170 micro-stepper and the DR60 DC brush drives.

Pin #	Name	Description
1	X_EN	X Enable. Active low. (OL= 0)
2	X_PWM0	X Command 0. PWM format 0-5V.
3	X_STL	X Settling. Dynamic breaking command. Active low.
4	Y_FLT	Y Fault feedback from drive. Active low.
5	Y_PWM1	Y Command 1. PWM format. 0-5V.
6	5VF	Field 5V.
7	RTNF	This is the common pin to all signals between the drive and the SB214ND.
8	-12VF	Field -12V.
9	Z_FLT	Z Fault feedback from drive. Active low.
10	Z_PWM1	Z Command 1. PWM format. 0-5V range.
11	T_EN	T Enable. Active low. (OL= 0)
12	T_PWM0	T Command 0. PWM format.
13	T_STL	T Settling. Dynamic breaking command. Active low.

Pin #	Name	Description
14	X_FLT	X Fault feedback from drive. Active low.
15	X_PWM1	X Command 1. PWM format. 0-5V range.
16	Y_EN	Y Enable. Active low. (OL= 0)
17	Y_PWM0	Y Command 0. PWM format. 0-5V range.
18	Y_STL	Y Settling. Dynamic breaking command. Active low.
19	+12VF	Field +12V.
20	SCRN	Shield.
21	Z_EN	Z Enable. Active low. (OL= 0)
22	Z_PWM0	Z Command 0. PWM format. 0-5V range.
23	Z_STL	Z Settling. Dynamic breaking command. Active low.
24	T_FLT	T Fault feedback from drive. Active low.
25	T_PWM1	T Command 1. PWM format. 0-5V range.

5.4.18 Extension Connector

Connector type: D-Type Male 9 pins.
Mating - D-Type Female 9 pins.

Serial Extension Interface to I/O modules.

The interface uses 3 differential signals: Clock, Data and Direction.

Pin #	Name	Description
1	COM	Common Ground
2	DIR+	Direction (Output)
3	DIR-	Inverted Direction (Output)
4	5V	5V, fused by 1.6A fuse
5	SCRN	Shield
6	CLOCK+	Clock (Output)
7	CLOCK-	Inverted Clock (Output)
8	DATA+	Data (Input/Output)
9	DATA-	Inverted Data (Input/Output)

6. COMMUNICATION

Communication with a host computer or a terminal is done via a RS232C or RS422 serial communication link.or via the host bus.

6.1 Serial Communication

6.1.1 Mode Of Operation

300,1200,4800, 9600 Baud rate,

NO PARITY,

1 START BIT,

8 DATA BITS,

1 STOP BIT.

6.1.2 Setting Controller's ID

When more than one controller are connected to the same serial link, each controller should be assigned an **ID** number. Each controller is shipped with the **ID** set to 0 as a default. Change the **ID** software parameter to the desired number and afterwards execute a **SAVE** command.

For example, to change the ID to 1: **SID1<cr>**

Using a controller whose **ID** is other than 0 requires to use the Wake command.

When power is switched on, the unit whose ID is 0 controls the communication link, and responds to communication requests. All the others are active, do respond to broadcast commands, but do not prompt back.

6.1.3 Baud Rate Setting

The serial communication **B**aud **R**ate is programmable via the **BR** parameter. The following rates are supported: 300, 1200, 2400, 4800, and 9600 (default). When the baud rate is changed, the new value becomes effective only after a **SAVE** command, switching power off, and re powering the unit.

SBR4800<cr>	Setting **BR** to 4800 bps.
SBR300<cr>	Setting **BR** to 300.
SBR 50<cr>	Wrong value. **BR**←9600. communication error 21 is prompted.

6.1.4 Partial Communication Shutdown

When the partial communication switch 1 is 'ON', only responses to ACSPL '**Input**' statements, and user defined function keys (FKEY_#) are accepted. Also, only messages that are generated by **ACSPL 'disp'** statements are transmitted. It is recommended to partially shut down the communication link when it is necessary to prevent unauthorized operations by the user.

(This feature is not supported by the SB214PC2)

6.2 SB214PC/PC2 Bus Communication

The SB214PC/PC2 communicate with the PC host via a two way FIFO with 8 bit data width, 1024 character deep.

At the end of each message that the controller writes into the transmit FIFO, it interrupts (IRQ) the host. See page 5-18.

6.2.1 Receive process

The receive FIFO is scanned by the controller every 0.001 second. If it is not empty, its content is read consecutively until an <End of message> character is reached, or until the FIFO is empty (RFE bit is set). The length of a message is limited to 254 characters excluding the <End of message> character. The received message is placed into a buffer and will be processed only after receiving of the <End of message> character.

6.2.2 Transmit Process

Messages are transmitted to the host in the following cases:

Prompts to commands that are issued by the host.

Responses to requests by the host.

Internally initiated messages of Begin & End motion, and Program termination.

ACSPL <disp> messages.

The controller tries to write the complete message to the transmit FIFO. If the FIFO is full, it waits about 1msec before writing the rest of the message into the FIFO. Immidiately after writing the <End of message> character, the controller activates the IRQ.

When the controller is set for bus communication, it also scans the serial link. So commands can also be issued via the serial link. This feature is a useful debugging tool. If there is no bus communication between the host and the controller, send **SFM<cr>** command via the serial link and the controller will switch the communication via the serial link only.

6.3 SB214VME Bus Communication

The SB214VME is a slave type VME compatible card. It communicates with the VME master controller via a two way FIFO with 8 bit data width, 1024 character deep.

At the end of each message that the controller writes into the transmit FIFO, it interrupts (IRQ) the VME master. The IRQ number and its level are programmable. When the transmit FIFO is full, the controller interrupts the VME master. The Vector Number (as seen by the H/W) reflects the interrupt source.

6.3.1 The Flag Register

The Flag Register reflects the status of the Recieve and Transmit FIFOs:

B7	B6	B5	B4	B3	B2	B1	B0
NU	NU	NU	NU	RFE	RFF	TFE	TFF

NU - Not Used

B0 TFF. Transmit FIFO is full
B1 TFE. Transmit FIFO is empty
B2 RFF. Receive FIFO is full
B3 RFE. Receive FIFO is empty

The content of the Flag Register is available also through the FR report parameter.

6.3.2 The Vector Number And The Q Array

The **V**ector **N**umber (**VN**) parameter contains the IRQ level and the vector number information. It is a 16 bit number - V15..V0.

VN15	VN14 - VN8	VN7 - VN2	VN1	VN0
0	IRQ LEVEL	VN NUMBER	0	0

VN1 - VN0 Forced to 0
VN7 - VN2 Vector Number. Range: [0 - 63]
VN14 - VN8 IRQ level. Only one bit must be set. Set bit 8 for level 1, set bit 9 for level 2 and so on up to level 7 (VN14 is set)
VN15 Forced to 0

The Vector Number can be changed automatically according to the message type. The content of the appropriate I entry of the Q array is automatically loaded into the VN before the interrupt is generated. The format of each Q element is identical to the format of VN.

Q(i)	Message Type
Q(0)	Controller’s prompt
Q(1)	Begin X message
Q(2)	End X message
Q(3)	Begin Y message
Q(4)	End Y message
Q(5)	Begin Z message
Q(6)	End Z message
Q(7)	Begin T message
Q(8)	End T message
Q(9)	Program message
Q(10)	System error message
Q(11)	Responses to Tell commands
Q(12)	Rsponse to Report commands
Q(13)	Response to Array Report commands
Q(14)	ACSPL. Program handling (list, compile, etc.)
Q(15)	ACSPL Disp statements
Q(16-19)	For future use

Upon power up, if the content of the Q array is legal, then it is used. If not, then according to the address setup by the address DIP switch, the following number is used:

SW3	VN
off	2040h
on	20A4h

The default value of Q is zero (illegal).

When the content of Q is illigal, the user can change the content of VN before each message.

The content of Q is saved in the non volatile memory. The content of VN is not saved.

6.3.3 Receive Process

The receive FIFO is scanned by the controller every 0.001 second. If it is not empty, its content is read consecutively until an <End of message> character is reached, or until the FIFO is empty (RFE bit is set). the length of a message is limited to 254 characters excluding the <End of message> character. The received message is placed into a buffer and will be processed only after receiving of the <End of message> character.

6.3.4 Transmit Process

Messages are transmitted to the host in the following cases:

Prompts to commands that are issued by the host.

Responses to requests by the host.

Internally initiated messages of Begin & End motion, and Program termination.

ACSPL <disp> messages.

The controller tries to write the complete message to the transmit FIFO. If the FIFO is full, it waits about 1msec before writing the rest of the message into the FIFO. Immidiately after writing the <End of message> character, the controller activates the IRQ using the proper IRQ level and vector number.

When the controller is set for bus communication, it also scans the serial link. So commands can also be issued via the serial link. This feature is a useful debugging tool. If there is no bus communication between the host and the controller, send **SFM<cr>** command via the serial link and the controller will switch the communication via the serial link only.

---

7. INPUT & OUTPUT PORTS

The following I/O are available as standard:

- 16 digital inputs
- 16 digital outputs
- 2 analog inputs10 bits resolution [0 - 5V].
- 4 analog inputs, 12 bit resolution,±10V. SB214PC (4 axis) only.
- Analog outputs. One per axis. 10 bits resolution. ±10V.
- One general purpose analog output. 8 bit resolution. ±10V.

7.1 Digital Inputs

The unit includes 16 general purpose inputs (in addition to safety inputs).

Some of the input also serve as registration inputs:

IN6 - X axis registration input.

IN7 - Y axis registration input.

IN9 - Z axis registration input.

IN10 - Z axis registration input.

When a registration input changes its state, the location of the axis is latched into the registration mark position parameter (**$M1**). If **AUTO_M1$** automatic routine exists, it is invoked and executed.

The **IN** parameter controls the polarity of the inputs. When a bit is 1, the corresponding input state is inverted by the software. The latching of position upon a registration input occurrence, and the corresponding automatic routines are not effected by the **IN** parameter.

When a **B**egin on **I**nput (**BXI, BXYI, BYZI etc.**) command is issued, the controller waits until the appropriate input is activated. The **IS** Input Source parameter defines which input is used. For example, if **XIS = 14**, then input 14 will be used upon **BXI**, **BXYI**, **BXZI**, and **BXYZI** commands.

Use the **RIP<cr>** command to monitor the status of each input.

Each input has a corresponding **ACSPL** state **- IN1, IN2,...IN16**.

Inputs 1 to 5 are supported by automatic routines **AUTO_IN1** to **AUTO_IN5**. See ACSPL for more details.

7.2 Digital Outputs

The 16 outputs can be used for general purpose as non dedicated outputs. The outputs can be assigned to the **PEG®** function, or to some other predefined function. The X axis PEG® uses outputs 1 to 8, and the Y axis PEG® uses outputs 9 to 16. The other dedicated functions are to signal the following motion states: **B/E motion**, **Ready**, **Early Ready** and **Intrpolation complete**.

B/E motion	Signals when the axis is in motion ("ON") or not in motion ("OFF"). X uses output 1. Y - output 2. Z - output 3. T - output 4.
Ready	After a **B**egin on **I**nput **(BXI),** it signals when the calculation are completed, and the axis is **READY** for motion. X uses output 5. Y - output 6. Z - output 7. T - output 8.
Early Ready	Signals that the axis is ready to accept a new move command ("**Early Ready**"). Also, when a super imposed move (**MD**) is executed in Master-Slave mode, this output is active. X uses output 9. Y - output 10. Z - output 11. T - output 12.
Interpolation Complete	Signals that the axis interpolation (profile generation) for the present move is completed ("**Intrp. Comp**."). X uses output 13. Y - output 14. Z - output 15. T - output 16.

The function of each output is controlled by the **O**utput **M**ask (**OM**) and by the **P**eg **M**ask (**PM**) parameters. When the appropriate **PM** bit is set, the corresponding output serves the **PEG®** function and cannot be changed by output handling commands. When the **PM** is cleared and the **OM** bit is set, the appropriate output serves the dedicated function as described above. When serving a dedicated function (not the **PEG®**), the outputs can still be set and reset by the **SHI, SLO** and **SOP** commands, (and by the **let HI, let LO**, and **let OP** commands in **ACSPL** program. See **ACS Software Guide** for more details).

To use outputs 1 and 2 for dedicated function:
SOM3<cr> - (3d =00000011b)
To watch the status of the output port type **ROP<cr>.**

7.3 Analog Inputs

there are two 10 bit analog inputs - AI1(ACSPL's A0) and AI2 (ACSPL's A1). They can be used as a joystick input or as general purpose analog inputs.

There are four additional 12 bit analog inputs only for the SB214PC 4 axis controller - AI3 to AI6. They are referred to by ACSPL as A2 to A5 respectively. See Analog / Joystick Inputs, page 5-65

7.4 Analog Output

One 10 bit analog output is available per axis - A_OUT. Its value is controlled by the $**D4** axis parameter. The output voltage range is -10V (D4=-511) to +9.99V (D4=511). It can be used also to monitor motion and other real time variables.

One 8 bit analog output is also available. It's value is controlled by the D0 parameter. The voltage range is -9.9V (D0=-127) to +9.9V (D0=127).

8. MOTION MONITORING

The actual velocity, position, position error, master position, and the current (= torque) commands can be monitored via the axis analog output **D4**. Each axis has its own dedicated analog output. It is a 10 bits Digital to Analog converter. The feature is controlled by three axis parameters - **MN**, **DC** and **MF**:

8.1 MoNitor ($MN)

MN=0	The axis analog output is controlled by the **D4** variable.
MN=1	Velocity signal.
MN=2	Drive's vector current (= torque) signal.
MN=3	The monitored parameter is defined by the bit assignment of the **DC** (Data Collection) parameter.

8.2 Data Collection ($DC)

Setting MN to 3 invokes a similar mechanism to Data Collection, and the monitored variable is defined by the bit content of DC. If more than one bit are set, then, the variable which is defined by the least significant bit is monitored.

DC bit assignment

7 -15	6	5	4	3	2	1	0
0	**A0**	**MP**	**0**	**DO**	**PE**	**CP**	**LV**

8.3 Monitor Factor ($MF)

The analog signal equals to the value of the parameter multiplied by 2^{MF}.

The output voltage is equal to:

$$(\text{Internal value}) \bullet 2^{MF} \bullet 10/512 \text{ [volts]}$$

To prevent an overflow, the setting of XMF must fulfill the following condition:

$$-511 < (\text{Internal value}) \bullet 2^{MF} < 511$$

8.4 Monitoring The Velocity Profile

The velocity scale factor depends on the value of D22. Internally, the maximum velocity value is represented by the value 1638. For example, if D22 was set according to a maximum velocity of 1,000,000 counts/second, then when the actual velocity is 1,000,000 counts/second, the internal (integer part of the) velocity value is 1638. When the actual velocity is 500,000 counts/second, the internal velocity value is 819.

To monitor the velocity set **MN** to 1. **MF** must be set to a value that will prevent overflow of the output. In the above example if the actual velocity is getting close to 1,000,000 counts/second (internal velocity of 1638, set **MF** to -2. If the actual velocity is less than 500,000 counts/second (internal velocity of 819) set **MF** to -1. If the actual velocity is 100,000 counts/second (internal velocity of 163) you may set **MF** to 1. This will provide a better dynamic range for the monitored signal.

8.5 Monitoring Position Error

Monitoring the position error is useful for improving the tracking quality, for finding the source of torque disturbances and setting the acceleration feed forward (**AF**).

To monitor the position error set **MN** to 3, and **DC** to 4. Usually the error value is less than 500, so **MF** should be set to a non negative value.
The following table provides the scale factor as function of **MF**:

MF	VPE [volts/count]
-2	0.00488
-1	0.00976
0	0.0195
1	0.039
2	0.078
3	0.156
4	0.312

8.6 Monitoring Current Position

Monitoring the position **CP** is a useful tool for measuring settling time. Usually the value of **CP** is large. While moving the monitor signal will overflow. Ignore it. Just examine the last part of it, where the axis settles. The [Volts/counts] scale factor is identical to the scale factor for position error monitoring.
For example, while moving in MM1 between **XCP=0** and **XCP=32768**, it is needed to see if the axis overshoots the target point by more than 10 counts. Set , **XDC** to 2, **XMN** to 3, and **XMF** to 3. A 10 counts overflow will generate a 1.5V overshoot at the end of the move.

8.7 Begin/End Signal

Another useful signal is the Begin/End Signal, available at the output port. To activate that feature, the Output Mask (OM) bits 1 to 4 must be set properly.
(See Input & Output Ports, page 7-1)

The timing accuracy of the signals is better than 1 msec.

9. INDEX

—F—

—G—

—H—

—I—

—J—

—K—

—L—

—T—

—U—

—V—

WARRANTY

The manufacturer warrants its hardware products against defects in material and workmanship for a period of one year from receipt by end user. During the warranty period, the manufacturer will, at his option, repair or replace products which prove to be defective.

The warranty applies to hardware only.

Exclusions

The above warranty shall not apply to defects resulting from improper or inadequate use, unauthorized modifications or misuse.

Limitations

Under no circumstances will the manufacturer be liable in any way to the user for damages, including any lose of profits or savings or other incidental or consequential damages arising from the use or inability to use the product.