ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Danaher Motion / IDC FP220* at our website: ***Click HERE***

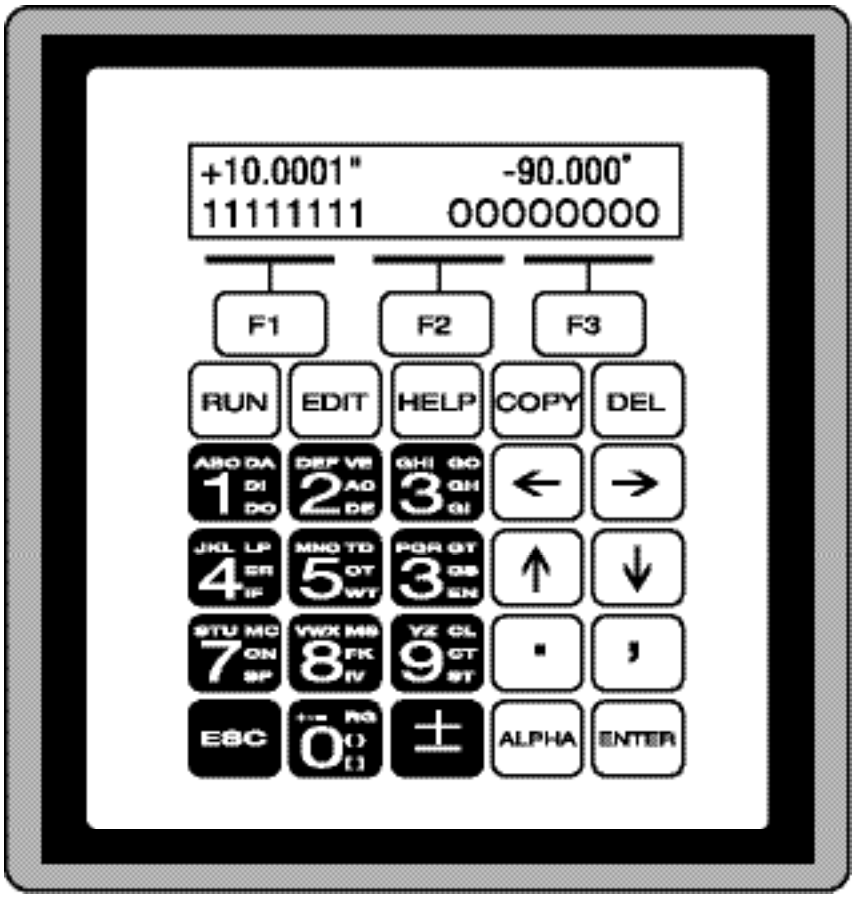
B8961 and B8962

Brushless Servo
Smart Drives

This manual covers the following IDC Products:
B8961-1 axis Brushless Servo Smart Drive
B8962-2 axis Brushless Servo Smart Drive

Operator's Manual

P/N PCW-4744, Rev 1.9



DANAHER MOTION

Table of Contents

PRODUCT AND MANUAL OVERVIEW 1

SHIPPING CONTENTS 2

QUICK START 3

KEYPAD OPERATION 7

KEYPAD FEATURES 7
Keypad Hardware Features 9
Dipswitches 9
Contrast 9
Remote Mounting 9
KEY PAD PROGRAMMING MENU STRUCTURE 10
RUN Menus 10
PROG Running Programs 11
JOG Jogging the Motor 11
TEST Test and Debugging functions 12
EDIT Menus 13
PROG Creating and Editing Motion Programs 13
SETUP System Set-up Menu 14
POS Reset Current Position to Zero 15
LIST List User Memory Usage 15
TUNING Servo Tuning 15
HELP Menus 16
At the Main Menu 16
At the Menus and Sub-Menus 16
In the Editor 16
COPY Menus 16
PROG Copy a Program to Another Program 16
TO PAD Upload Memory to Keypad 17
FROM Download Memory to Control 17
DEL Menu 17
RUN-TIME OPERATOR INTERFACE 17

CONFIGURING YOUR SYSTEM 19

SETUP MENU 20
MOTOR Motor/Drive Configuration 20
ENC Encoder Configuration 22
MECH Mechanical Configuration 24
I/O Input & Output Configuration 28
JOG Jogging Configuration 36
HOME Homing Configuration 37
PROG Program-Run Configuration 39
RS232 RS-232C Configuration 40
MISC Miscellaneous Configuration Parameters 40
TUNING YOUR SERVO SYSTEM 43
Compensator 44
RS-232C Users 44
Key Pad Tuning 45
Keypad Tuning Menu Tree 45
GAINS Setting the Servo Gains 45

Toggle Tuning Stimulus....46
Monitor Monitor Port....46

PROGRAMMING YOUR APPLICATION....47

SMART DRIVE PROGRAMMING OVERVIEW....47
CREATING OR EDITING PROGRAMS WITH THE KEYPAD....47
COMMAND SUMMARY....48
VARIABLES AND ARITHMETIC....49
VARIABLES....49
LEGAL VERIABLE NAMES....49
BUILT-IN VARIABLES....50
NON-VOLATILE VARIABLES....52
ARITHMETIC OPERANDS AND EQUATIONS....53
BOOLEAN OPERATORS- & (AND), | (OR)....53
LOGICAL OPERATIONS ON EXPRESSIONS....54
INCREMENTING AND DECREMENTING VARIABLES....54
EXPRESSIONS....54
MULTI-AXIS OPERATION....55
SIMPLE GO COMMANDS....55
TYPICAL PROGRAMMING EXAMPLES....58
READING AN ANALOG INPUT....59
CONFIGURING AN ANALOG OUTPUT....60

PROGRAMMING COMMANDS....61

PROGRAMMING WITH SERIAL COMMUNICATION....91

SECTION 1:RS-232 PROTOCOL....91
Making RS-232C Connections to the S6961/2....91
Troubleshooting Serial Command Problems....92
Daisy Chaining Smart Drives....92
SECTION 2:APPLICATION DEVELOPER SOFTWARE....93
Installing Application Developer....93
Common Installation Errors and Remedies....94
SECTION 3:USING APPLICATION DEVELOPER™....95
Setup Wizard....95
Axis Setup....99
I/O Setup....102
Miscellaneous (Misc) Setup....103
File Menu....104
Program Editor....105
View Configuration....106
Communications....106
Run Menu....107
SECTION 4:RS-232C IDEAL™ COMMAND REFERENCE....108
Overview....108
Sample File....108
Command Syntax....110
Serial Setup Commands....112
Serial Programming Commands....118
Commands Not Used In Host Mode....120
Serial Immediate Status Commands....122
Serial Supervisory Commands....127

HARDWARE REFERENCE 131

B8961 AND B8962 SMART DRIVE MOUNTING.....131
FK1 and FK2 Fan Kit 132
RPACK-1 and RP ACK-2 Mounting Installation 132
Remote Keypad Mounting 132
B8000 SMART DRIVE HARDWARE SPECIFICATIONS.....133
B8961 AND B8962 HARDWARE CONNECTIONS 134
B8961 AND B8962 INPUT AND OUTPUT SCHEMATICS 135

TROUBLE SHOOTING 137

SYMPTOMS AND SOLUTIONS 137

IDC PRODUCT SUPPORT 139

FACTORY AUTHORIZED DISTRIBUTORS 139
REGIONAL OFFICES.....139
TOLL FREE TECHNICAL SUPPORT.....139
CAD LIBRARY.....139

WARRANTY & REPAIRS.....139

INDEX.....140

APPENDIX A: IDC ACTUATOR RATIOS.....143

CONFIGURING INCH & MM UNITS ON SMART DRIVES USED WITH IDC ACTUATORS.....143
Methods For Configuring Ratio 143
IDC ACTUATOR RATIOS.....143
N, T, R2, R3, R4, NM, RM Series 143

APPENDIX B: EMC INSTALLATION 147

ELECTROMAGNETIC COMPATIBILITY (EMC) INSTALLATION GUIDELINES.....147
Introduction 147
Separation and Shielding 148
Grounding.....149
Physical Installation 150
AC Power Line Filtering 150
Motor Output Power Cable Shielding 151
Declaration of Conformity 152

APPENDIX C: FAN KIT OPTION 153

APPENDIX D: REGEN PACK OPTION 155

Product and Manual Overview

IDC's B8961 and B8962,one and two axis Smart Drives,combine a DSP based servo amplifier with a powerful,but easy to use,machine controller.We offer a single axis version (B8961) and a dual axis version (B8962).The B8962 has two axes of independently configurable amplifiers.Both products incorporate IDC's 5 amp continuous,10 amp peak,120/230VAC,digital servo amplifier with a motion/machine controller in a single,compact package.Up to 24 digital I/O are available to control other machine functions.Up to 8 of these can be configured as analog inputs or outputs.

The B8000 Series Smart Drives feature a 5 amp continuous, 10 amp peak, digital brushless servo drive using state of the art DSP technology to provide high performance closed loop servo control to a wide variety of permanent magnet brushless and brushed servo motors and actuators offered by Industrial Devices Corporation.

The Vector Control used by the servo amplifier offers higher bandwidth torque and velocity control when compared with sinusoidal or hall effect commutated brushless servo amplifiers.This results in an expanded torque speed curve that leads to higher throughput machines.The easy servo setup offered by the keypad configuration or via Application De veloper gives high speed,high torque,brushless servo motor performance,with an ease of use rivaled only by a stepper system.

IDC's IDeal™ Smart Drives provide excellent value when your application calls for any of the following:

- A small,1-2 axis machine controller
- A "motion node"to a master PLC or PC
- A cost effective operator interface
- Thumbwheel data inputs
- Flexible I/O capabilities
- On-board OPTO I/O modules (analog or digital)

The B8000 Series Smart Drives also feature a programming/operator interface.The intent of the Smart Drive's keypad is to allow you to quickly and easily configure and program your system.Initial setup and configuration,in most cases,will require only a few minutes using the keypad.While Smart Drives fully support PC based RS232 programming,the setup and configuration sections of this manual often emphasize the keypad approach.RS232 commands and setup definitions are in a separate chapter.We strongly recommend using a keypad with your first B8000 Series Smart Drive application. You will also find it an invaluable troubleshooting and machine commissioning aid during subsequent applications.

The following manual has been designed to help you successfully install,program and operate your IDeal Programmable Smart Drive System.Each section of the manual emphasizes the common theme "Ease of Use".If you have any questions that this manual does not adequately answer, please contact our factory application engineers.

The manual begins with this Overview, followed by a Quick Star t chapter to help you quickly confirm basic system operation.Next the manual describes Communication with Your Control via the keypad interface.RS-232 interface details are covered in Chapter 7, RS-232 Operation .

Chapter 4 covers the procedure for Configuring Your System to your specific equipment and application requirements. It includes step-by-step keypad instructions on scrolling through and entering setup parameters. This chapter covers initial motor settings, I/O configuration, and defining the mechanics of your system. IDC's Windows based Application Developer also follows the same menu structure described here. .

Chapter 5, Programming Your Control , provides detailed program and application examples and strategies. Other topics include variable usage, user menus, math functions, and analog I/O. Our IDEAL™ command language is generally regarded as the easiest motion control language in the industry. It is both easy to remember and intuitive, without sacrificing flexibility or power.

In Chapter 6, Command Reference, this alphabetical listing of Smart Drive commands explains in detail syntax, ranges, defaults, and provides programming examples.

Chapter 7, RS-232 Operation , is for users who plan to use the Smart Drive in an RS232 hosted mode, or who wish to program and configure the drive via RS232. IDC's Application Developer program follows a standard Windows dialog box structure to make configuring and programming the Smart Drive control straight forward. This section also covers RS232 command syntax and definition for users who are not using Windows.

The Hardware Reference chapter provides detailed I/O schematics, as well as motor drawings, cables and specifications. The Troubleshooting chapter lists a series of common application problems along with their symptoms and solutions. Included with this manual is the Application Developer cd-rom. IDC's Application Developer and Servo Tuner software are automatically installed on your hard drive by running the setup program on the Application Developer cd-rom. This disc also includes a readme file containing the latest information on software features. Many demo programs are included with AppDeveloper.

If you cannot find the information that you need in this manual, please contact the Applications Department. (800) 227-1066.

Shipping Contents

1. B8961 or B8962 Smart Drive control
2. 120 VAC power cable
3. Application Developer software cd-rom
4. Installed OPTO modules (optional)
5. Fan Kit (optional)

Note: Units ordered as B8961NPs or B8962NP do not include the following keypad items:

1. Keypad programming/operator interface (Part #FP220)
2. Keypad remote extension cable (6 ft)
3. Keypad remote mounting gasket

Quick Start

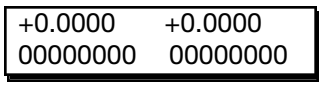
The purpose of the Quick Start chapter is to help an experienced motion control user quickly set-up and bench test a B8000 Series Smart Drive with an IDC supplied motor/encoder package.The following directions assume that the user is familiar with servo motors,encoders,servo amplifiers,servo controls,and their related electrical connections.Please refer to Chapter 4, "Configuring Your System" if you have questions on any of these procedures.

For ease of set-up, we recommend that you use a keypad, and in this chapter it is assumed that a keypad is used for set-up, even if an RS232 connection will be used later.

Caution: All connectors are KEYED to prevent "backwards" installation. The connectors should fit snugly, but do not need to be forced.

Caution: Do not plug the motor/encoder connector into the drive until instructed. The motor and encoder can be damaged by an incorrect motor setting.

Apply Power

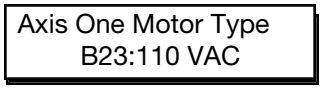


Connect the B8000 Series Smart Drive to 120VAC using the supplied AC power cable and connector. The AC power input terminals are labeled and located at the top of the Smart Drive.When power is applied,the display briefly shows Model number and Software Revision,then changes to the Main Display.The Main Display continuously shows the position of each axis on the top line and the status of Inputs 1-8 and Outputs 1-8 on the bottom line.

Select the Motor Type



1. Press the EDIT key. Press SETUP (F2),then MOTOR (F1),then TYPE (F1).
2. The display should show NONE.Use the ↑ and ↓ keys to change the motor choice.If you have a B8962,use the ← and → arrows to change to axis two to verify the motor choice.



3. Once the proper motor and voltage level has been selected,press ESC to save your choices.
4. You will be prompted to Enable each axis.Choose YES (F1).

If using a non-IDC motor, please call the factory for details on obtaining a motor data file for your motor. If you already have this custom motor data file, see the Using Non-IDC Motor section of Chapter 7, RS-232 Operation .

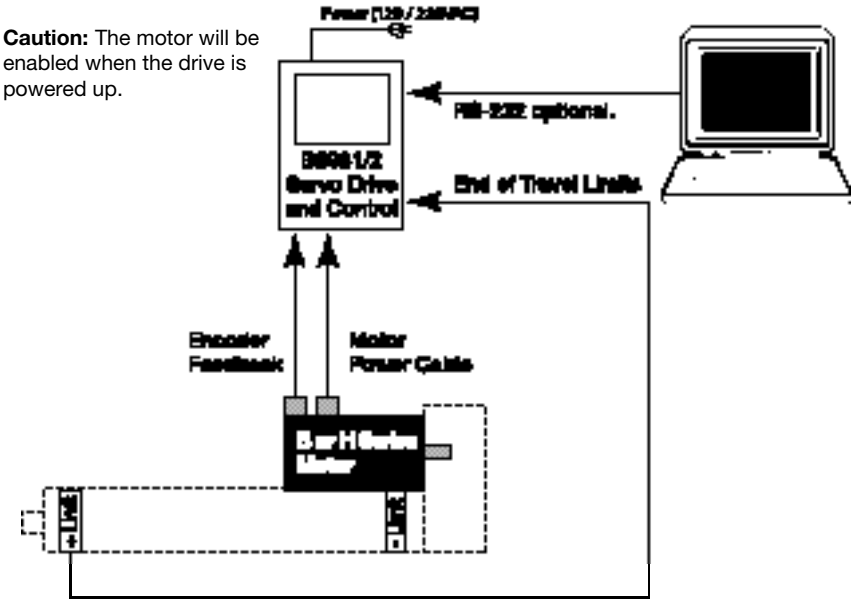
Connect The Motor and Encoder Cable(s)

Power down the drive and connect the pre-wired motor and encoder cables to the Smart Drive.The 20 pin motor/encoder cable plugs into the bottom of the Smart Drive.The MS connectors are keyed at the motor end.See the diagram below for more information.If necessary, please refer to the Hardware Reference chapter for wiring diagrams.

End-of-travel inputs are jumpered at the factory. See the Hardware Reference section for IDC supplied limit switch wiring schematics.

After all motor and encoder cables are firmly connected, reapply power and continue on with the Quick Start directions.

Caution: The motor will be enabled when the drive is powered up.



Jog the Motor

1. Press RUN. Press JOG (F2).
2. Press the ← or → key to jog (and ↑ and ↓ keys to jog axis two on the B8962).
3. Change the Jog Speed by pressing High (F2) or Low (F1) and jog again.

```
------ RUN ------
PROG   JOG   TEST
```

Run [TEST MOVE] Program

Press ESC to return to the Main Display.

1. Press RUN. Press TEST(F3).Press MOVE(F3)
2. Select the axis to test.(F1,F2,or F3) The axis selected will move forward and backward one distance unit.(Distance units default to motor revolutions.)

```
------ TEST MOVE ------
Axis 1   Both   Axis 2
```

Create and Run a Simple Program

Press ESC to return to the Main Display.

1. Press EDIT. Press PROG (F1). Enter 1.(Edit program #1.)
   AC1 Press F2 & 2 & 1 & ENTER
   VE10 Press F1 & 2 & 1 & 0 & ENTER
   DI3 Press F2 & 1 & 3 & ENTER
   GO Press F1 & 3 & ENTER
   EN Press F3 & 6 & ENTER
2. Press ESC to save the program and return to the Main Display.
3. Press RUN, then PROG (F1),then enter 1.(Run program #1.) The program should move the motor 3 revolutions.

Discrete Input Test

Connect a 12" test wire to one of the COM screw terminals found on the side of the drive.One by one,touch the test lead to each Input screw terminal and verify that the input status shown on the Main Display toggles between 0 and 1.

Discrete Output Test

1. Press RUN. Press TEST (F3). Press OUTPUT (F2)
2. Press the ← or → key to select an output number.
3. Press the ↓ or ↑ to turn the output ON or OFF.
4. Press ESC to leave this menu.The outputs will return to their state prior to entering the test mode.Please use caution when manually toggling outputs connected to live devices.No software interlocks are observed during this test.

The OPTO position's default configuration is as an input.Please see Configuring Your System for information on changing these to Outputs.You will then be able to toggle their state from this screen.

Where to go from here?

Successfully completing these steps confirms the basic operation of your system.At this point, you may want to refer to the following sections for more information:

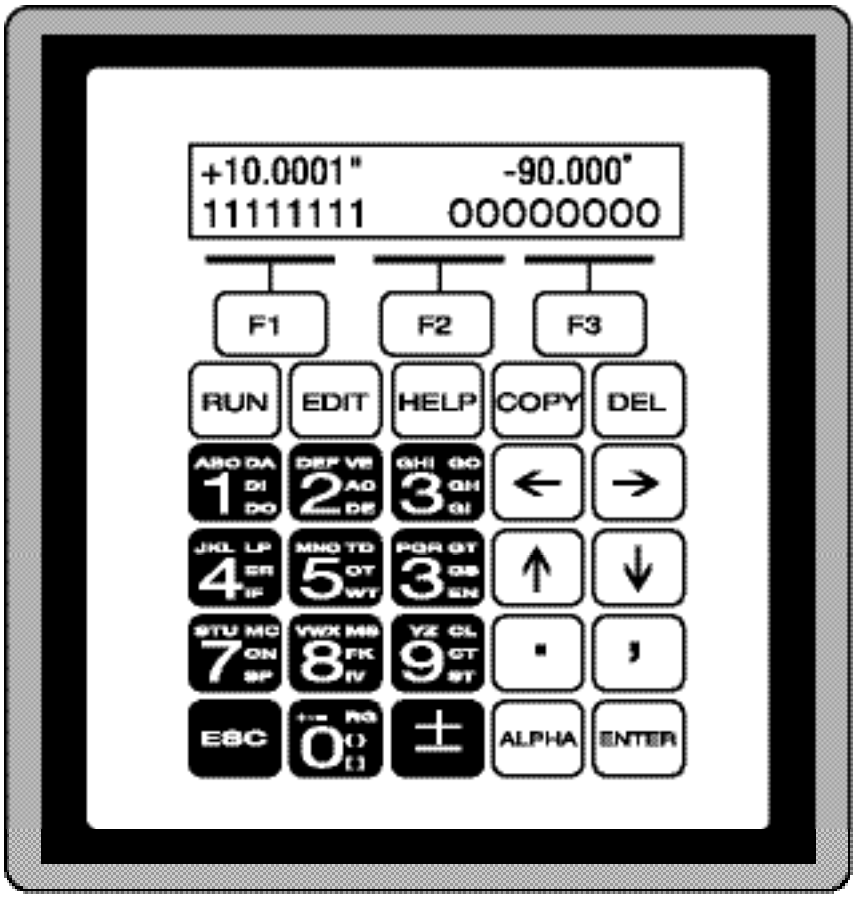
- Hardware connections p.134
- Configuring Your System p.19
- IDC Actuator Gear Ratios p.25
- Non-IDC's motor operation p.21
- Drive mounting p.131
- Fan Kit and RPACK connections p.132
- I/O software configuration p.28
- I/O hardware schematics p.135
- Basic keypad operation p. 7
- Programming examples p.58
- RS232 connection and usage p.91

Keypad Operation

This chapter is meant to familiarize a first time user with the basics of IDC's keypad operation.IDC's keypad functions fall into two categories. The keypad functions first as a programming and troubleshooting tool.Secondly, it functions as an operator interface.Operators can run programs,make menu choices,or be prompted to enter data via the keypad number keys.

The first section of this chapter, Keypad Features , defines the basic function of each button on the keypad.The second section, Menu Structur e, gives the programmer a broad overview of how the keypad's setup and programming menus operate and fit together. Detailed information about each step parameter is presented in Configuring Your Control .

Keypad Features



Display

Easy to read,2 line,40 character, back-lit supertwist display.

F1, F2, F3 Function Keys

Used as Menu selectors.Used with numeric keys to select commands in the editor. Programmable as operator menu selections.See the FK command for information on using the function keys within a program.

RUN

Runs a Program, Jogs an axis,or accesses Test/Debugging functions like Program Trace mode,and amplifier Enable/Disable/Reset.

EDIT

Edits setup parameters,programs,tuning (servos only),and resets position counter. Also accesses program listing and teach mode. (Teach mode will be implemented in a future software r evision.)

HELP

Provides help on keys,menus and command syntax.

COPY

Copies one program to another within a unit.Copying a complete application, including all programs,setup parameters and motor configuration information. Smart Drive to Smart Drive copying,via the keypad,will be implemented in a future software release.

DEL

Deletes characters in the editor, or entire programs from memory.

Arrows (←, →, and ↑, ↓)

Scroll through menu options,setup choices and programs in the editor. Also used to move an axis in jog mode.

Decimal Point

Used to enter fixed point numbers.

Comma

Used in multi-axis controls to separate axis command parameters.Also part of the syntax in message and variable prompt commands.

ALPHA

Alpha plus a numeric key selects the first letter on that key. Press the numeric key more than once to select second or third characters. For example,Alpha+1+1 selects B.Other ASCII symbols,such as the >< and ↑ characters can be selected with Alpha using the ↑ and ↓ arrow keys.

ENTER

Selects a choice and enters a space in the editor.

Sign (+/-)

Selects the direction of motion in programs and can be used in math expressions.

ESC

Stops a program,backs up a menu level,also used for exiting and saving programs while in the editor.

Numeric Keys (0-9)

Enters numbers.Used with Alpha to select characters.Used with F1-F3 to enter commands in the editor.

Keypad Hardware Features

Dipswitches

A bank of four dipswitches on the back of the keypad provide a way to lock operators out of various keypad access levels.See the chart below for switch assignments.If access to a menu level is denied,pressing that key will have no effect. For example,if 1 is on,and 2 is off, the operator will be able to stop motion, and program execution,(by pressing the escape key) but will not be able to access the RUN menus to select another program to run.This is a hardware inhibit,and is independent of any software,or setup parameter in the Smart Drive. Power must be cycled before keypad dipswitch changes take effect.

In addition,access to the jog menu can be enabled or disabled from software.

SW1	SW2	**Function Level**
off	off	Full keypad functionality
off	on	No access to RUN, ESC,EDIT, COPY, DEL menus
on	off	No access to RUN, EDIT, COPY, DEL menus
on	on	No access to EDIT, COPY, DEL menus

Switch 3 and 4 are reserved for future functionality.

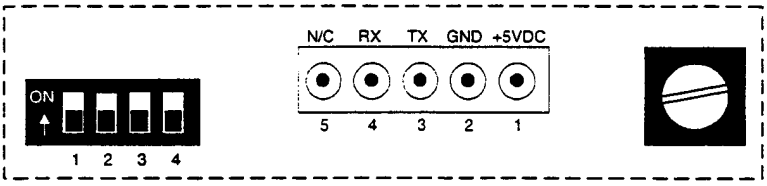
Contrast

On the back of the keypad,there is a single,plastic, flat head screw driver adjustable,potentiometer.This is used to adjust the contrast on the LCD display. If the Smart Drive and keypad were purchased together, this adjustment has been made by IDC.Some adjustment may still be needed to accommodate unusual lighting or viewing angles.

Remote Mounting

The keypad can easily be remote mounted and sealed to NEMA 4 specifications by using the included mounting gasket and 6 foot communication cable.The gasket should be installed with its adhesive side to the enclosure.A pressure seal is formed between the gasket and the keypad,while the adhesive maintains the seal between the enclosure and the gasket.The keypad communicates with the Smart Drive via RS-232 so this cable can be extended if needed.At longer distances,users may have to provide a separate 5 VDC supply to power the keypad.Complete keypad drawings and pinouts can be found in the Hardware Reference chapter.

Back of Keypad Detail drawing



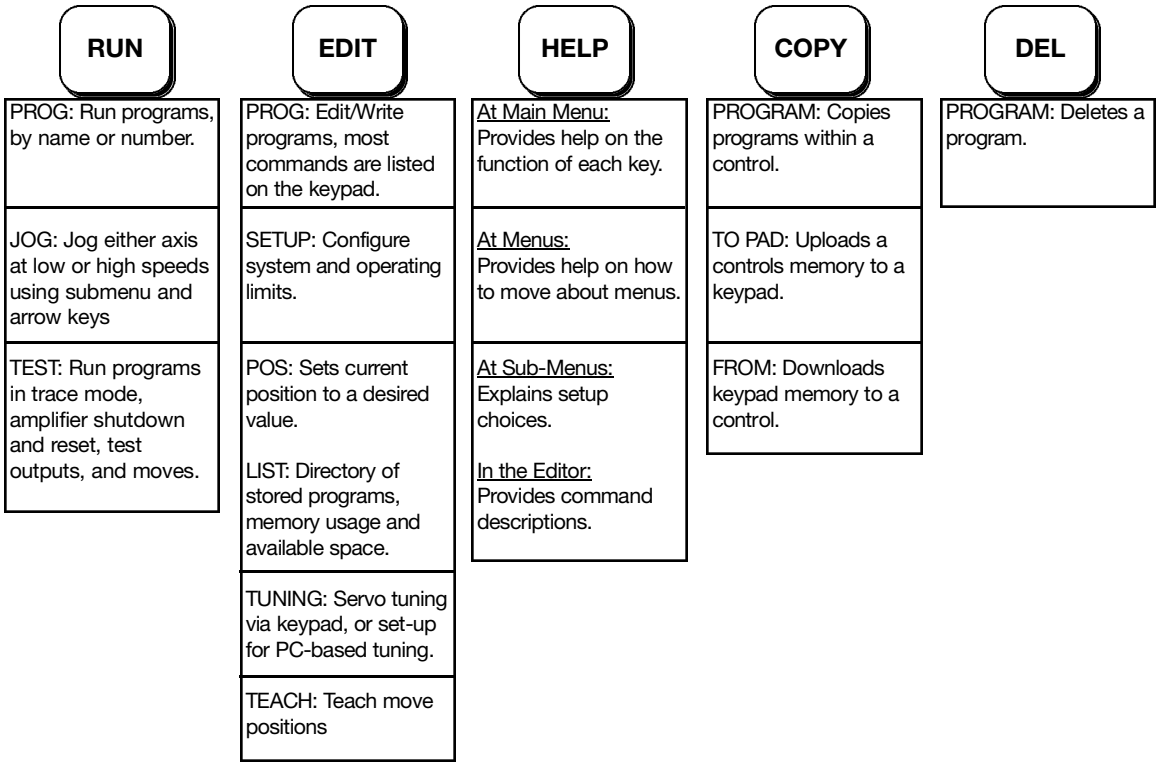
Key Pad Programming Menu Structure

Most operations from the Keypad are menu-driven.A menu consists of a title bar on the top display line and as many as three options (or sub-menus) at a time on the bottom display line.Each option is positioned over one of the function keys. Pressing a function key will select the corresponding option.

If a menu has more than three options,arrows will appear in the title bar indicating that there are more options which are not displayed.Pressing the appropriate arrows (indicated on the display) will cycle through all options associated with the menu. To leave a menu,without making a selection,or to back up one menu level,press ESC.

The following tree shows the Menus which are accessible from the Main Display by pressing RUN, EDIT, HELP, COPY, and DEL keys.

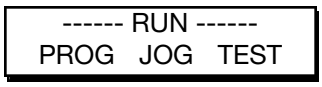
NOTE:ESC backs up one menu in SETUP, and returns the user to the Main Display elsewhere.



RUN Menus



Pressing the RUN key displays a second set of RUN sub-menus.Access the sub-menus by pressing the function keys F1,F2,and F3,which are below PROG, JOG and TEST respectively.



PROG Running Programs

Run an existing program by program number, by pressing:

```
↑ RUN PROGRAM ↓
> 5
```

1. PROG(F1)
2. Program number 1-199 using numeric keys
   (1-400 available with the 30k expanded memory option)
3. ENTER

Run an existing program by name by pressing:

```
--↑ RUN PROGRAM ↓--
> 12 GRIND
```

1. PROG(F1)
2. ↑ and ↓ keys to scroll through the list of available programs until you find the program you want
3. ENTER

JOG Jogging the Motor

Jog either axes by pressing:

```
JOG AXIS 1 +0.0000
<LO>    HIGH
```

1. RUN
2. JOG(F2)
3. ← and → keys to move axis 1
   ↑ and ↓ keys to move axis 2 on a B8962.

Note:The ↑ and ↓ keys will also jog Axis 1 on a single axis unit.

Change between Low and High speeds with the F1 and F2 keys. Jog speeds and accelerations can be changed in the JOG \ SETUP menu.

"Jog"an incremental distance by:

```
JOG AXIS 1 +0.0000
Dist.: .01
```

1. RUN
2. JOG (F2)
3. Entering a number (i.e.,0.012).
4. Pressing and releasing an arrow key will make the motor move this distance. The arrow pressed determines the direction of the move.
5. Repeat 3 and 4 until at desired position.

Additional pushes of arrow keys will jog the same distance until either, F1,F2,or F3 is pressed.This feature is intended for very fine, final positioning.The incremental jog speed is therefore fixed at a very low speed.

TEST Test and Debugging functions

TRACE

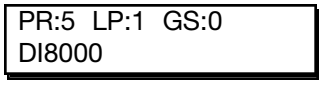
The trace features allows you to de-bug programs by sequentially executing one program command at a time.



Press:

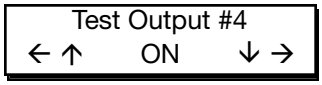
1. RUN \ TEST \ TRACE
2. Enter the program name or number
3. ENTER

The top line displays the program number, the number of nested loops,and the number of nested routines.The bottom line shows the command to be executed when you press ENTER.Each time you press ENTER,the displayed command will be executed.Pressing ESC halts program execution.TRACE mode is not currently supported during homing operations.



OUTPUT

This feature allows you to test the control's outputs, as well as the devices to which they are connected, by forcing them on and off. Press:



1. RUN \ TEST \ OUTPUT
2. ← → keys to scroll through outputs 1-8,and any OPTO positions configured as outputs.
3. ↑ ↓ keys to turn the output on and off

Please use caution when connected to live devices.The Outputs will revert to their original state when ESC is pressed.

NOTE:The OPTO position default to Inputs.They must be configured as outputs from the SETUP / I/O / OPTO menu before being accessible from this test utility. See Configuring Your System for details.

MOVE

This selection moves your motor shaft one user unit forward and backwards.This allows you to verify basic motor, encoder, and amplifier operation.

Press: RUN \ TEST \ MOVE,and select the axis to move.Pressing the axis Function key will start the move.

SHUTDN

Selecting SHUTDN (shut down) allows you to enable,disable or reset axis 1 or axis 2 (B8962). When a drive is disabled,the amplifier is off and your motor has no power.The shaft can be manually rotated relatively easily. RESET returns each drive to its power-on condition. Existing set-up parameters are restored.



Press:

1. RUN \ TEST \ → \ SHUTDN
2. ENABLE,DISABLE or RESET

RS232 Serial Communications

RUN > TEXT > RS232

```
---- Test Connection-----
TRANSMIT   RECEIVE
```

This feature allows for testing of the terminal serial communications port through the keypad.

Testing Serial Transmission:

1. From the Test Connection menu press the F1 key to select the TRANSMIT option

```
Test String 'ABC123'
Transmitting....
```

2. The SmartDrive will now transmit the string "ABC123" every 5 seconds.

Testing Serial Receive:

1. From the Test Connection menu press the F3 key to select the RECEIVE option.

```
Data Received
```

2.Any character received on the terminal port will be displayed on the keypad.

ENCODER

To be implemented in a future version of software.

EDIT Menus

EDIT

Pressing the EDIT key reveals three sub-menus called PROG, SETUP and POS.

```
------ ↑ EDIT ↓ ------
PROG   SETUP   POS
```

Pressing the ↓ key reveals three more EDIT sub-menus called LIST, TUNING and TEACH.

```
------ ↑ EDIT ↓ ------
LIST  TUNING  TEACH
```

Access the PROG, SETUP, POS, LIST, TUNING and TEACH sub-menus by pressing the appropriate function key. See descriptions of each of these sub-menus below.

PROG Creating and Editing Motion Programs

This menu allows you to edit an existing program, or begin entering a new program from the keypad.

```
-↑ EDIT PROGRAM ↓-
-> 5
```

Use the numeric keys to enter a program number to start a new motion program, or use the ↑ and ↓ keys to scroll through the list of existing programs, and press ENTER.

You are now viewing your IDeal™ motion program in the keypad program editor and are ready to edit your program.

```
AC3 VE1 DI8000 GO
EN
```

See Programming Your Application for more information on editing IDeal motion programs with the keypad.

SETUP System Set-up Menu

The following table shows the structure within the SETUP menu. For complete descriptions of each system parameter in the SETUP sub-menu,see Configuring Your System .

Sub-Menus	Set-up Parameters	Description of Set-up Parameters
MOTOR	TYPE	Motor type
	D-RES	Drive Resolution
	DIR	Direction of travel
ENC	MODE	Select open/closed loop mode
	E-RES	Encoder resolution (fixed at 8000 on IDC motors)
	FOL-ERR	Following error
MECH	DIST	Distance Units
	RATIO	Scale distance to preferred user units
	BKLASH	Electronic backlash compensation
	VEL	Speed units
	VMAX	Critical speed limit
	ACCEL	Acceleration units
I/O	INPUTS	Input functions
	OUTPUTS	Output functions
	OPTOS	OPTO module configuration
JOG	ACCEL	Jog acceleration
	LO-VEL	Low jog velocity
	HI-VEL	High jog velocity
	ENABLE	Enable/disable jog in RUN menu
HOME	EDGE	Edge of home switch
	SWITCH	Type of home switch
	OFFSET	Position counter offset
	FINAL	Final homing direction
PROG	PWR-UP	Program to run on power up, if any
	SCAN	How to scan program select inputs
	DELAY	Program Select de-bounce time
RS232	ECHO	Echo characters
	UNIT#	Serial address
MISC	DISPLAY	Display mode
	STOP-RATE	Decel rate when stop input activated
	TEST	Enable Test Menu

POS Reset Current Position to Zero

POS is a quick way to reset the motor's current position to (absolute) zero,which can be a helpful setup and debugging tool.
Press:

Reset Position?
YES NO

1. EDIT \ POS
2. YES (F1) or NO (F3)

LIST List User Memory Usage

LIST provides a way to view your program memory usage.Your IDC Smart Drive will accept up to 199 IDeal™ motion programs of 1024 characters each,with 6K bytes of program storage available (up to 400 programs available with the 30k expanded memory option).

Pressing:

1. EDIT \ ↓ \ LIST
   Displays the number of programs stored in your Smart Drive.

DIRECTORY ↑MORE↓
PROGRAMS: 18

2. ↓
   Displays the total amount of memory your programs have used.

DIRECTORY ↑MORE↓
BYTES USED: 1186

3. ↓
   Displays the number of bytes of memory you still have available.

DIRECTORY ↑MORE↓
BYTES FREE: 4958

4. ↓↓↓ ...↓
   Displays the number of bytes being used by each of your programs.

DIRECTORY ↑MORE↓
5 <untitled>: 56 bytes

TUNING Servo Tuning

The following shows the structure of the TUNING menu.See Configuring Your System for a complete description of servo tuning and the parameters in the TUNING sub-menus.

Sub-Menus	Set-up Parameters	Description of Set-up Parameters
AXIS 1	GAINS	Alter servo gains for AXIS 1.
	TOGGLE	Start step function generator to view system response.
	MONITOR	Configure oscilloscope monitor port.
	NOHUNT	Enable between-move Anti-Hunt feature.
	WINDUP	Enable Inertial Anti-Windup feature.
AXIS 2		Tune Axis 2 (same set up parameters as AXIS 1 above)
RS232		Configure Smart Drive for tuning via RS-232C port and IDC's Servo Tuner

TEACH Teach Move Positions Within a Program

To be implemented in a future version of software.

HELP Menus

HELP

Whenever you have a question using the keypad,pressing HELP will display a help message related to the menu you are currently in.Help messages are often several lines,which you can scroll through using the ↑ and ↓ keys.When you are finished reading a help message,pressing ESC returns you to the menu.

At the Main Menu

```
- - - - ↑ HELP ↓ - - - -
Use RUN key to...
```

HELP explains the functions available when you press any of the non-numeric keys.

At the Menus and Sub-Menus

```
This option is used to
select the motor type...
```

HELP explains the selections available from your current menu location.

In the Editor

```
↑ COMMAND SUMMARY ↓
AC Acceleration
```

HELP provides a brief alphabetical command list.Full syntax and details on command usage are available in the IDeal™ Command Reference of this manual,or from HELP in the Application Developer editor.

COPY Menus

COPY

Copying programs from one name (or number) to another can save you a significant amount of time when programming your Smart Drive.The keypad has its own non-volatile memory, which allows you to upload the memory (motion programs,set-up and tuning parameters) of one Smart Drive, reconnect and download the memory contents to another unit.Note:Smart Drive to Smart Drive copying via the keypad is not yet implemented in the B8000 Series of Smart Drives.

Pressing the COPY key brings up three choices that can be accessed by pressing the function keys.

```
- - - - - COPY - - - - -
PROG   TO PAD   FROM
```

PROG Copy a Program to Another Program

This selection allows you to copy any existing program to a new program name.

```
↑ SOURCE PROGRAM ↓
> 5
```

1. Press PROG
2. Enter the source program number. Or, if you wish, you can scroll through your list of program names by using the ↓↑ keys.
3. Press ENTER

Then you are asked to enter the new program.If the target program already exists, you will have to delete it first (see DEL).

Then

```
↑ TARGET PROGRAM ↓
> 5
```

1. Enter the target program.
2. Press ENTER

Remember to change the name of the copied programs to avoid subroutine call conflicts.

TO-FROM PAD Copy to-from Keypad

```
----------COPY----------
PROG  TO PAD  FROM
```

The COPY TO-FROM feature has been implemented which allows user setup and programs to be down-loaded to and from the keypad.A special keypad cable (PCS-5004) provides a +5V power supply and a 9 pin D style connector for communications with Application Developer™.COPY TO-FROM requires SmartDrive version v6.00 or higher and keypad firmware v2.60 or higher. Contact IDC for firmware upgrades and cable information

Copy TO Keypad:

1.Press F2 key to select TO PAD (Note:Unit address is not saved in keypad)

Copy FROM Keypad:

1.Press F3 key to select FROM (Note:Unit address is not set from keypad transfer)

In order to use COPY TO-FROM with Application Developer™, connect the kepad to the PC using cable PCS-5004 and select “Retrieve All”from the Application Developer Communications menu to load data from the keypad and select “Send All”from the Application Developer Communications menu to load data to the keypad.See the Application Developer section in the S696X,B896X and 96X manuals for more information on using Application Developer.

DEL Menu




```
↑ DELETE PROGRAM ↓
> _
```

The DEL key lets you delete any motion program currently in your Smart Drive.
Press:

1. DEL
2. Enter the number of the program to delete.Or, if you wish, you can scroll through a list of existing program names by using the ↓↑ keys.
3. ENTER

Run-Time Operator Interface

Your keypad features an easy-to-read,2-line,40-character, back-lit display that can be easily remote mounted to a panel and exposed to industrial NEMA 4 environments.

While motion programming and system configuration procedures using the keypad are defined by IDC, run-time operation (i.e.how the machine operator will interface with the Smart Drive) falls completely within the customer’s control.

Here are some of the operating functions you can program with the Smart Drive:

- Run a program on power-up,on input signal from a PLC,or RS-232C host command
- Within a program,prompt the operator for any program variable (the number of parts to run,size of parts,speed,etc.)
- Run a part or program by name
- Lock-out operators from programming functions

For more information on programming your Smart Drive’s operator interface,see the Programming Your Application and IDeal™ Command Reference chapters.

Configuring Your System

This chapter presents a straight forward,intuitive procedure for Configuring Your System to your specific equipment and application requirements.The following configuration section contains detailed,step-by-step directions for customizing the B8961 and B8962 to your specific application and mechanical requirements.IDC recommends that even experienced users follow this procedure in its entirety. Following all the SETUP steps will insure that no critical parameters are overlooked.

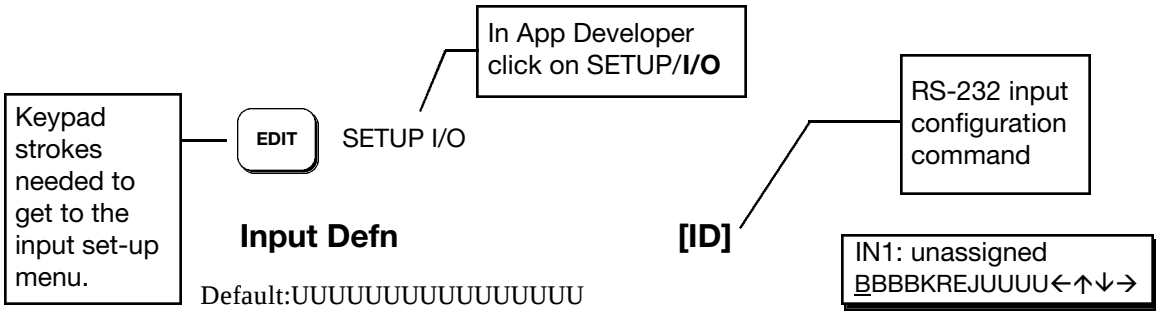
The task of configuring your B8961 or B8962 Smart Drive system to a specific application consists of customizing a number of software parameters to match the mechanics of the system.These parameters include distance,acceleration and velocity scaling as well as I/O configuration.Other than plugging in the motor/encoder cable and connecting your I/O wiring,there are no hardware settings on a B8961 or B8962 Smart Drive system.

All of the software configuration can be done via the keypad,or via RS232 using Application De veloper.The Configuring Your System description presented here is from a keypad user's perspective,via the keypad menu structure and step-by-step keypad instructions.

Application De veloper and RS-232C users will want to refer to this chapter for detailed explanations of configuration parameters.Details on how to use IDC's Application De veloper can be found in the Chapter 7. For RS-232C terminal users, non-Windows™ PC users,or PLC users,the equivalent 2-character ASCII configuration commands are detailed in the RS-232 Command Reference section Chapter 7,but the examples presented here are valid for all users.The 2-character ASCII command appears in brackets next to the appropriate keypad menu choice in this chapter. Application De veloper users will find that the Windows dialog boxes under the **Setup** menu follow the keypad menu structure very closely.

Users who plan to create and download their own ASCII setup file will also find this section useful.The RS-232 command that relates to the parameter being set is in brackets below the keypad menu structure.RS-232 command syntax details are in the RS-232 Command Reference section of Chapter 7.

Each SETUP step follows the format of the example below:



The function of each input is easily configured using the k eypad as described on the fol - lowing page.

SETUP Menu

Pressing the EDIT key followed by the SETUP function key reveals three menus.
Pressing ↓ reveals a second and a third set of menus.
Each of these sub-menus are explained below.







NEW MOTOR CONFIGURATION



The structure of the motor type selection menu has changed in order to accommodate the addition of the built-in linear servo motor files plust the expansion of the B servo motor series.The MOTOR TYPE menu option now has three sub-menus: STEPER (for step motor drive parameters), R-SRVO (for rotary servo motors) and L-SRVO (for linear servo motors).

Configuring Stepper Motor Type

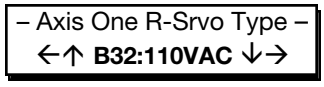
EDIT > SETUP > MOTOR > TYPE > **STEPER**



Default:	STEPPER	(S696X)
	N/A	(B896X)
	INDEXER	(96X)

Configuring R-SRVO Motor Type [MT]

EDIT > SETUP > MOTOR > TYPE > **R-SRVO**



Default:	N/A	(S696X)
	NONE	(B896X)
	INDEXER	(96X)

This option specifies the type of rotary servo motor connected to the B896X SmartDrive and the operating voltage level.The motor parameters used for drive configuration have been specifically tailored for IDC supplied rotary servo motors.

1.Use the ← and → keys to select an axis.
2.Use the ↑ and ↓ keys to scroll through the list of IDC's rotary motors
3.Press the ESC key to select
None (Sets K_{ii} and K_{ip} gains to zero.)
B12:110VAC
B22:110VAC,B22:220VAC
B23:110VAC,B23:220VAC
B23H:110VAC,B23H:220VAC
B32:110VAC,B32:220VAC
B33:110VAC,B33:220VAC
B40:110VAC,B40:220VAC
B41:110VAC,B41:220VAC
B42:110VAC,B42:220VAC
BN21:110VAC
BN23:100VAC
BN31:110VAC
BN32:110VAC
H3:110VAC
H4:110VAC
or OTHER
Default:None.

Consult the factory if you intend to use a non-IDC motor. See Chapter 7, RS-232 Operation, for information on configuring a Smart Drive for a non-IDC motor.

CAUTION: If the motor type is set incorrectly the motor may be damaged. Specifying the incorrect voltage will not damage anything, but different current loop gains are used so system instability may result.

The safest way to change motor types is to:
1. Power up the control without the motor connected.
2. Change the Motor Type setup parameter to the new motor.
3. Remove power.
4. Connect the motor.
5. Re-apply power.

EDIT SETUP MOTOR-**D-RES**

- Axis One Drive Res -
←↑ 8000 counts/rev ↓→

Motor Resolution [MR]

Default:B Motors = 8,000,H motors = 8,000,BN Motors = 4,000.

IDC's B8961 and B8962 Smart Drives allow you to set a different drive input resolution than the actual resolution encoder.This option specifies the input resolution for each axis of your digital servo smart drive,in counts per motor revolution.This is a very useful feature when retrofitting microstepping systems.

It is very important that the encoder resolution (E-RES) is set to your post-quadrature encoder resolution.D-RES can be set at the users discretion.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ and ENTER keys to select from a list of drive resolutions available: 200,400,1000,2000,5000,8000,10,000,18,000,25,000,25,400,and 36000.

Your mechanics,and application needs will dictate the choice of D-RES. For example,if you want to move in .1 degree increments,a D-RES choice of 18,000 will allow 50 motor steps per degree and prevent any resolution induced rounding errors.

EDIT SETUP MOTOR **DIR**

- Axis One Motor Dir -
←↑ POSITIVE ↓→

Motor Direction [MD]

Default: Positive

This option provides a convenient way to change which direction the motor moves when you program a positive distance command.

When POSITIVE is selected as the motor direction,the EOT+ limit switch should be wired so that moves in the plus direction (as shown on the keypad display, or via the PA command) will activate the switch.When NEGATIVE is selected,the EOT+ limit switch should be wired so that moves in the negative direction (as shown on the keypad display, or via the PA command) will activate the switch.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ and ENTER keys to select a direction

ENC Encoder Configuration

EDIT SETUP ENC **MODE**

- Axis One Enc Mode -
←↑ CLOSED LOOP ↓→

Encoder Mode [EM]

Default: SERVO CLOSED LOOP

This option sets the encoder mode for each axis.

The B8961 and B8962 servo position loops are closed by the DSP in the amplifier, freeing up the motion control board to do other processing.Therefore, you can run the servo in either “open loop”or “servo closed loop”modes.This allows you to view (on the keypad display or via the PAn commands) either the commanded or actual position in user units.All programmed moves are based on the Motor/Drive Resolution setting (D_RES).

1. Use ← and → keys to select an axis.
2. Use the ↑ and ↓ keys to scroll through the list of encoder modes and press ESC to select:

Encoder Mode	Description
OPEN LOOP	The OPEN LOOP position will be displayed on the keypad.
OPEN-STALL	The OPEN LOOP position will be displayed on the keypad but the encoder will be used for stall detection.
CLOSED LOOP	The encoder position is displayed on the keypad.All moves are based on encoder position and stall detection is enabled.
SERVO-CLOSED	The encoder position is displayed on the keypad.All moves are based on the commanded OPEN LOOP position and stall detection is enabled.
CLOSED LOOP PM	The encoder position is displayed on the keypad.All moves are based on encoder position,however, post move correction algorithms will keep the encoder position equal to the last commanded OPEN LOOP position. Following error is still active while in CLOSED LOOP PM mode.A following error will occur when the number of correction steps exceeds the following error value.This allows the SmartDrive to signal a fault when the displacement can not be corrected (i.e.the motor is stalled at an obstruction). Position maintenance will not attempt to correct position while navigating menus with the keypad.

EDIT > SETUP > ENC > IN-RANGE > **WINDOW**

- Axis One PM DeadBnd -
←↑ 25 Steps ↓→

Configuring Position Maintenance Deadband [IR]

Default: 25 motor steps
Range: 0 - 99999

Position maintenance deadband is a user definable region surrounding the commanded position in which the motor shaft can reside and not be considered “out of position”.A displacement position exceeding the last commanded position +/- the deadband value,will cause position maintenance to attempt to correct the position.

1. Use ← and → keys to select an axis.
2. Use numeric keys to enter a new deadband value and press ENTER then ESC to register.

NOTE: EDIT > SETUP > ENC > INRANGE > WINDOW and the IR serial command have an alternate functionality with servo SmartDrive systems. See the B8961/2 manual for more details.

EDIT > SETUP > ENC > **PMGAIN**

```
- Axis One    PMGain -
←   10   →
```

Configuring Position Maintenance Gain [PG]

Default: 10
Range: 1 - 32767

The position maintenance gain value is an integer factor usedin determining the velocity at which the position maintenance correction move will travel.Correction velocity is calculated as (displacement * correction gain) in units of steps/sec. Therefore,the larger the displacement,the faster position maintenance will attempt to correct position. For example,if the correction gain is set to 3 and an active displacement of 3200 steps occurs,the correction velocity will be (3 * 3200) = 9600 steps/sec.

1. Use ← and → keys to select an axis.
2. Use numeric keys to enter a new position maintenance gain and press ENTER then ESC to register.

EDIT > SETUP > ENC > **PMMAX**

```
- Axis One    PMMax -
←  1 RPS  →
```

Configuring Position Max Velocity [PV]

Default: 1 RPS
Range: 0.005 - 9999999

The position maintenance maximum velocity value sets a velocity limit in which position maintenance will attempt to correct position.Regardless of the magnitude of displacement or correction gain,the correction velocity will never exceed the maximum velocity setting.

1. Use ← and → keys to select an axis.
2. Use numeric keys to enter a new position maintenance max velocity in the same units selceted in teh SETUP>MECH>VEL menu and press ENTER the ESC to register.

EDIT SETUP ENC **E-RES**

```
- Axis One Encoder Res
←↑ 4000 counts/rev ↓→
```

Encoder Resolution [ER]

Default:8,000 for B motors,2,000 for H motors,4000 for BN Motors.

This option is used to set the encoder resolution for each axis.The resolution is specified in encoder pulses per revolution of the motor, post-quadrature.

Specifying the wrong encoder resolution will prevent the Smart Drive from properly communicating the motor and could damage it,as well as cause unstable system behavior. For this reason,IDC fixes the encoder resolution for our servo motors.Consult the factory if your application requires using another motor or resolution encoder.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ and ENTER keys to select from a list of encoder resolutions,or use the numeric keys to enter the encoder resolution.

EDIT SETUP ENC **FOL-ERR**

```
- - Axis One Fol Error - -
←      8000 steps      →
```

Following Error Limit [FE]

Default:8,000 counts

This option defines the maximum position following error allowed during motion. A fault occurs when the error between the commanded and feedback signal exceeds the Following Error value.
Range:0-524,000 step counts,0 = OFF

If a Following - Error occurs,the control will enter a fault state where:
- Any motion or program being executed is immediately terminated.
- The LCD Display will indicate “Following Error”,along with an explanation.
- A fault output will be generated if defined as a “Stall”or Fault output.
- The ESC key or a RESET (either an input or via RS-232) will need to be issued before motion can occur.

MECH Mechanical Configuration

Through the mechanical setup menu, your IDeal™ Programmable Smart Drive allows you to program in distance, velocity and acceleration units convenient for your application.Once configured, your keypad will use these units in all display and position reporting modes.This menu also allows you to compensate for a known amount of backlash in your mechanical system,and to set a maximum allowable speed for each axis.

```
-- ↑ MECH SETUP ↓ --
DIST  RATIO  BKLASH
```

Pressing the MECH function key displays three menu choices:

```
-- ↑ MECH SETUP ↓ --
VEL   VMAX   ACCEL
```

Pressing ↓ reveals three additional menu choices:

EDIT SETUP MECH **DIST**

- Axis One Dist Units -
←↑ revs ↓→

Distance Unit [DU]

Default:Motor Revs

DIST is used along with RATIO to select your distance units and unit label.All distance values specified in the system will be expressed in the units selected here. The relationship between motor revolutions,system mechanics,and the distance label chosen here is defined with the RATIO command defined below.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ keys to select distance units from:
   - mils
   - inch
   - feet
   - yards
   - mm
   - cm
   - meter
   - arcsec
   - arcmin
   - degrees
   - radians
   - grads
   - steps
   - %
   - index
   - revs

Notes:

- You can change DIST or RATIO at any time.Changing them will not change the associated DI or DA values in a program.(i.e.DI100 will command a 100 inch move instead of a 100 step move if the DIST units are changed from Steps to Inches.)
- If steps is chosen,the control automatically fixes the RATIO (see below).

EDIT SETUP MECH **RATIO**

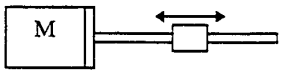
- - Axis One Ratio - -
←↑ 1 to 1 ↓→

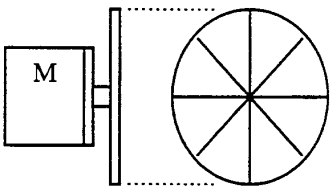
Gear Ratio [GR]

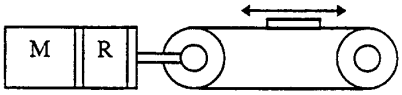
Default:1 to 1

The RATIO option is used to scale DI and DA moves to your preferred distance units.RATIO sets the ratio of motor revolutions per DIST unit . Up to 5 digits on either side of the ratio can be entered to properly scale your DIST units.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a ratio expressed as 2 integers.Ex:When programming in output shaft revolutions of a 5:1 gearbox,enter “5 to 1” rather than “1 to .2”







Notes:

- You can change DIST or RATIO at any time.Changing them will not change the associated DI or DA values in a program, so all moves will change by the same factor that RATIO was changed.
- If using an IDC supplied actuator, the proper Gear Ratios for programming in units of Inches and mm can be found in Appendix A,directly following the Index.

Units Example - Lead Screw System
• Desired distance units:inches
• Lead Screw:4 revs/inch
DIST = inch
RATIO = 4 to 1

Units Example - Rotary Index Table
• Desired distance units:1/8 of a revolution
DIST = inch
RATIO = 1 to 8

Units Example - Gear Reduced Tangential Drive System
• Desired distance units:mm
• Reducer:5:1 reduction
Drive Pulley:6 inch circumference
5 revolutions of motor travel results in 152.4 mm of linear road travel.This ratio must be expressed as an integer to be used in the Gear Ratio command.Multiply each side by 10 to get a Gear Ratio of 50 to 1524.
DIST = mm
RATIO = 50 to 1524

EDIT SETUP MECH **BKLASH**

```
- - Axis One Backlash -
←   25.0   steps   →
```

Backlash [BK]

Default:0.0 (Distance Units)
(Not yet implemented)

BKLASH allows your Smart Drive to compensate for the backlash in each axis of your mechanical system.The Smart Drive will automatically take up (add) the backlash distance specified whenever the move direction is reversed.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a backlash value in DIST units.

EDIT SETUP MECH **VEL**

Velocity Units [VU]

Default:rps (motor Revolutions Per Second)

```
- Axis One Vel. Units -
←↑   mm/sec   ↓→
```

Use this option to select your velocity units.All velocity values specified in the system will be expressed in these units.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ and ENTER keys to select velocity units from the list:
   - rps
   - rpm
   - (DIST units)/sec (see DIST above)
   - (DIST units)/min (see DIST above)

EDIT SETUP MECH **VMAX**

Maximum Velocity [MV]

```
- - Axis One Max Vel. -
←↑  50.0 inch/sec  ↓→
```

Default:50 (velocity units)

This parameter limits the top speed of your motor. Depending on the application, you may want to limit the speed of your control to prevent accidental damage to your mechanics. For example,in a leadscrew driven system, exceeding the "critical speed"will damage the leadscrew or ballscrew.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to set the maximum velocity in VEL units.

EDIT SETUP MECH **ACCEL**

Acceleration Units [AU]

```
- Axis One Accel. Units
←↑        sec        ↓→
```

Default:sec

This option is used to select acceleration (and deceleration) units.All acceleration and deceleration values specified in the system will be expressed in these units. You can specify acceleration as a rate,or in time-to-accelerate to full speed.

1. Use ← and → keys to select an axis.
2. Use the ↑↓ and ENTER keys to select acceleration units from the list:
   - sec (time to reach top speed)
   - (DIST units)/sec^2
   - rps^2 (motor Revolutions Per Sec^2)

EDIT > SETUP > MECH > **AMAX**

```
– Axis One Max Accel –
←  0.002 sec  →
```

Configuring Acceleration Maximum [AM]

Default: 0.002 (acceleration units)
Range: 0.002 - 99999999 (acceleration units)

Acceleration maximum sets a maximum acceleration and deceleration limit for programmed move profiles in the current acceleration units.Programmed accelerations and decelerations for moves will be limited by this parameter (analogous to VMAX for velocity).Regardless of acceleration units,the absolute maximum acceleration is 0.002 seconds.

1. Use the ← and → keys to select an axis.
2. Use numeric keys to enter a new acceleration maximum in the same units selected in the SETUP > MECH > ACCEL menu and press ENTER then ESC to register.

I/O Input & Output Configuration

Your IDeal™ Programmable Smart Drive has eight discrete optically-isolated inputs, and eight discrete optically-isolated outputs.It also supports up to eight Opto I/O modules (G4/G5 footprint),which is like having a G4PB8 board and cable built in — plus,it has the capability of supporting a mix of analog and digital modules.All the inputs and outputs can be configured to a specific machine control function.

The eight OPTO positions can be inputs or outputs,logic or AC/DC power, digital or analog,or even used as a thermocouple (Type K or J) inputs.The control is completely protected from errors in interchanging modules. You will not damage the unit by plugging in the wrong module.Simply insert the modules you need into the positions you desire,and configure each position in the OPTO menu (explained below) as either an input or an output.

Your IDeal Programmable Smart Drive is compatible with almost any manufacturer's G4 or G5 digital opto modules (OPTO 22,Grayhill,Gordos,etc.).However, at the time of this printing,only Grayhill's analog modules are compatible with our control.Other manufacturer's analog opto modules do not fit into a G4 footprint.

For more information on how to use your Smart Drive's inputs and outputs in an application, refer to Chapter 5, Applying the Product , and Chapter 6, Programming Command Reference , in this manual.

The function of each input and output in your system is easily configured in the I/O SETUP menus. Once you have your I/O defined,it is a good idea to write down your configuration scheme for later reference when you develop your motion programs.

```
- - - ↓ I/O SETUP ↑- - -
INPUT  OUTPUT  OPTOS
```

```
- - - ↓ I/O SETUP ↑- - -
    OUTSTS  LIMITS
```

```
 OUTPUT  STATES  ON
PWR-UP  FAULT  ST/K
```

OUTSTS (Output States on Event) allows user configuration of output states after power-up, fault or a Stop/Kill.LIMITS allows user configuration of the EOT (End of Travel) switch polarity. Also a new configurable input CLR CMD BUFFER "c"has been added.

EDIT SETUP I/O **INPUTS**

Input Definition [ID]

```
IN1: EXTEND JOG 1
EBBBKREJUUUU←↑↓→
```

Default:UUUUUUUUUUUUUUUU

The function of each input is easily configured using the keypad as described below.

The function for each input channel is indicated by a letter along the bottom of the display.The first 8 letters are for the dedicated Inputs and the last 8 letters are for the optional OPTO inputs.

OPTO positions configured as outputs are shown as dashes and cannot be configured without changing the position to an input in the OPTO menu.The OPTO position's default configurations are as inputs.(see below).

1. Use ← and → keys to select an Input.The function of the highlighted input will be displayed on the top line.
2. Once your cursor is on the desired input,use ↑↓ to select from the following list of dedicated functions for each input:

Character	Function
B	Binary Formatted Program Select
C	BCD Formatted Program Select
c	Clear Command Buffer
D	Disable Keypad
E, e	Extend Jog, AXIS 1, axis 2
F,f	Set CL Force
H	Z axis home input
I	Interrupt (Run 98)
J, j	Jog Speed, AXIS 1, axis 2
K	Kill
M, m	Motor Shutdown AXIS 1, axis 2
N	Analog Input
P	Pause/Continue
R, r	Retract Jog, AXIS 1, axis 2
S	Stop
U	Unassigned
V	Data Valid
W	Warm Boot
Z	Z Axis Fault

B Binary Program Select Input

Allows programs to be run remotely using a PLC, switches or outputs from a computer. Up to 199 programs may be selected using binary inputs.The lowest numbered input becomes the least significant selection bit (i.e.,input #1 is less significant than input #2).The act of configuring an input as a program select input also enables binary program select mode.

C BCD Program Select Input

Allows programs to be run remotely using a TM99 Thumbwheel module,PLC, switches or outputs from a computer. Up to 99 programs may be selected using BCD inputs.The lowest numbered input becomes the least significant selection bit (i.e.,input #1 is less significant than input #2).
The act of configuring an input as a program select input also enables the BCD program select mode.

c Clear Command Buffer

Clears the terminal input buffer and buffered commmand buffer.

D Disable Keypad

When activated,the keypad is disabled allowing NO user access.The keypad resumes normal operation,subject to the dipswitch pattern,when the input is released.

E, e Extend Jog (E specifies AXIS 1, e specifies axis 2)

When activated,the motor will Jog in the Extend (+) direction.When the input is released,motion stops at the Jog Accel rate.If an End of Travel limit is hit while jogging,the motor will stop at the Stop Rate.(see Edit-Setup-Misc.) Before the motor can be moved back off the limit,a Stop or Kill input must be activated to clear the fault generated by hitting an End of Limit switch.Alternatively, an **S** or **K** command sent over RS-232 will also clear the fault.

The velocity is determined by the Jog Speed Input and the Jog Low and High setup parameters.When the input is off the speed is low, and vice versa.If none of the inputs are configured for Jog Speed,the motor will jog at the Jog Low setting.

F,f Set CL Force (F specifies axis 1, f specifies axis 2)
In addition to being able to set a clamping current via the CL command,the user can also set the clamp curent on the fly based on an input.When the SET CL FORCE input is asserted during a CL defined move,the control will maintain the torque the motor is producing at that instant. For example,if the following move is executing:VE1 CL3 DA3 GO (CL units are Amps),and a SET CL FORCE input is activated when the motor current control is at 1.5 amps,the current will be clamped to 1.5 amps.If this input is not triggered,the control will clamp the current to 3 amps.The SET CL FORCE input requires a valid CL move to be in progress otherwise the input is ignored.The SET CL FORCE input is intended for applications clamping to load cell feedback.

H Z axis Home Input
Not yet implemented.

I Interrupt (Run 98)
When activated,motion on all axes is stopped,at the stop-rate (see Edit-Setup-Misc-Stop-Rate).The current program is stopped,and processing continues with the first command in program 98.If no program is running when the input is activated, program 98 will run.This input is ignored while the keypad is in Edit mode.This is a positive edge sensitive input, rather than a level sensitive input.If multiple inputs are configured as Interrupts,only the first edge of the first activated input will be seen.If subsequent Interrupt inputs go active while the first Interrupt input is active,no additional interrupts will be seen.

Advanced Interrupt handling can be achieved using the (INT98CRTL) and (ARM INT98) variables.The (INT98CTRL) variable determines whether Interrupts can be disabed or not.The (ARM INT98) variable allows you to arm and disarm the Interrupt as desired.

When the Indexer powers up (INT98CTRL) is initialized to 0.In this mode, evry interrupt results in an immediate jump to program 98, even if you just entered program 98.This means that it is possible for the interrupt service routine to be interrrupted by another interrupt input.This functionality is backwards compatible with earlier versions of firmware in IDC SmartDrives.The value of (ARM INT98) is ignored.

When (INT98CTRL) = 1 you can enable and disable Interrupts at will with the (ARM INT98) variable.Setting (INT98CTRL) = 1 also initializes (ARM INT98) to 1. This means the control is watching for interrupts.When (INT98CTRL) is set to 1 an interrupt causes the program to jump to program 98 AND sets (ARM INT98) = 0, disabling any further interrupts until you reenable them by setting (ARM INT98) = 1.This allows you to control when you want to reenable interrupts in your interrupt service routine (program 98).

To summarize,when (INT98CTRL)=1:

If (ARM INT98) = 0
Interrupts are ignored. (ARM INT98) is automatically set to 0 by the Smart Drive on the first edge of the input, if the previous (ARM INT98) value was 1. Interrupt processing will be suspended until (ARM INT98) is reset to 1.This allows for input debouncing and gives teh user control over the ability of program 98 to interrupt itself.

If (ARM INT98) = 1
The system is awaiting the firt INT98 input assert edge.Once the interrupt is seen the control will go to program 98.Subsequent interrupts are ignored until (ARM INT98) is set to 1.

(INT98CTRL) and (ARM INT98) are reset to default values on power-up.
Note: There is a space in (ARM INT98).

J Jog Speed (J specifies AXIS 1, j specifies axis 2)
This input works along with the Extend Jog and Retract Jog.When a jog input is activated the control checks the state of this input to determine the jog speed.If the input is OFF, the system will jog at the Jog Low speed.If it is ON it will jog at the Jog High speed.If the input is not configured the jog inputs will result in motion at the low speed.

K Kill Motion
When activated,causes the control to abruptly stop commanding further motion and terminates program execution.No deceleration ramp is used.Caution should be used because of the damage instantaneous deceleration could cause to your mechanical system.The Stop input provides a more controlled halt.(Implemented in V2.22 or later)

M, m Motor Shutdown: (M specifies axis 1, m specifies axis 2)
This input,only active at the top menu level,while a program is not running, disables the motor. All current on the motor will be removed, resulting in little or no holding torque.

N aNalog
Only the OPTO I/O may be configured as analog inputs.To use an analog or temperature I/O module the I/O position must be configured as an analog input signal.This tells the Smart Drive that the input is no longer a discrete input and prevents the input signal from being misinterpreted.

Analog signals are read into input variables (AI9) through (AI16) corresponding to OPTO positions 9 through 16.See Programming Your Application for details on using analog I/O.

P Pause/Continue
While this input is grounded program execution is stopped.Moves are not interrupted when the Pause input goes active.Command execution will Pause at the end of the move,and Continue when the input goes high.See the **ST** command for interrupting moves in progress.

R, r Retract Jog (R specifies axis 1, r specifies axis 2)
When activated,the motor will Jog in the Retract (-) direction.When the input is released,motion stops at the Jog Accel rate.If an End of Travel limit is hit while jogging,the motor will stop at the Stop Rate.(see Edit-Setup-Misc.) Before the motor can be moved back off the limit,a Stop or Kill input must be activated to clear the fault generated by hitting an End of Limit switch.Alternatively, an **S** or **K** command sent over RS-232 will also clear the fault.

The velocity is determined by the Jog Speed Input and the Jog Low and High setup parameters.When the input is off the speed is low, and vice versa.If none of the inputs are configured for Jog Speed,the motor will jog at the Jog low setting.

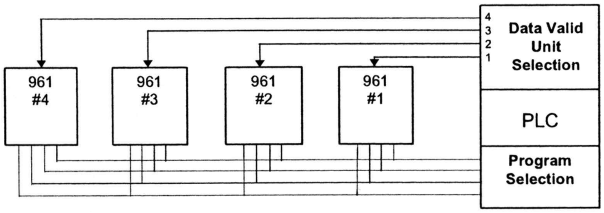
S Stop
When activated,any program execution or functional operation is immediately stopped.This includes any motion,time delays,loops,and faults.Moves will be decelerated at the stop deceleration rate.New programs will not execute until the stop input goes inactive.

See the **SCAN** setup parameter for more information on stopping program execution.See the **ST** command for more information on stopping moves without halting command execution.

U Unassigned
An Unassigned input functions as a programmable input,and can be used in IF and WT statements just like any of the dedicated function inputs.

V Data Valid
When the input is configured,it determines if the Binary/BCD program select lines are processed or ignored.If the input is active,program select lines are processed, otherwise they are ignored.This allows applications to be wired in a pseudo-bus architecture fashion with each unit sharing the same program select lines,and the data valid inputs determining which units should listen.Configuring this output can greatly reduce panel wiring.

In the example shown below, use the Data Valid input reduced the number of wires required for selecting programs by one-half.



W Warm Boot - System Reset
Resets the Smart Drive, clearing the RAM Buffer, and resetting the control to its power up state.Programs and setup parameters are not erased.This is typically used to restart system when a fault condition occurs.The power up program,if defined,will be run.

Z Z Axis Fault
Not yet implemented.

EDIT SETUP I/O **OUTPUTS**

Output Definition [OD]

OUT1: Programmable
PPPPPPPP - - - - ←↑↓→

Default:PPPPPPPP - - - - - - - -

The function for each output channel is indicated by a letter along the bottom of the display.The first 8 letters are for the dedicated Outputs and the last 8 letters are for Opto channels configured as outputs.Opto positions configured as inputs are shown as dashes and cannot be configured without changing the OPTO position to an output.

1. Use ← and → keys to select an Output or Opto output channel.The function of the highlighted output will be displayed on the top line.
2. Once your cursor is on the desired output,use ↑↓ to select from a list of function configurations for each channel.See next page.

Character	Function
A	Amplifier Fault
B, b	Brake
C	over Current
D, d	Direction
F	Fault
H, h	at Home
I	Indexer Step (Z axis)
K,k	At CL Limit
L	Limit Error
M, m	Move Complete
N	Analog Output
P	Programmable
S	Stall
T	Torque Mode
U	Unstable
Z	Z Direction

A Amplifier Fault
Output goes low on any amplifier fault.An amplifier fault may be due to temperature,motor short circuits, excessive following error, over-voltage and excessive regeneration conditions.

B, b Brake (B specifies AXIS 1,b specifies axis 2)
It is often advisable that applications using a ballscrew type actuator with a vertical load use a brake to prevent the load from falling in the event of a fault.Brake outputs will go on whenever a fault occurs and prevent a load from moving.

C Over Current / Over Voltage Fault
Not yet implemented.

D, d Direction (D specifies axis 1,d specifies axis 2)
The Direction Output indicates the direction of motion for a given axis.The output remains set until motion begins in the reverse direction.

F Fault
The fault output acts as a fail-safe fault output.Under normal operation the output is grounded (ON) and goes high (OFF) when any type of fault occurs.(Smart Drive software prior to V2.10 had this polarity reversed.) A fault can occur from any amplifier fault condition as well as for the following general faults:
- BMA (Board Monitor Alarm) time-out.
- Error finding Home.Both limits were hit.

The exact cause of the fault can be determined in a number of ways:
- Shown on the keypad's display.
- Over RS232 using the **SS**, **SD**, and **SA** status commands.
- Other outputs can be configured to show more specific fault states.

H, h At Home (H specifies AXIS 1,h specifies axis 2)
The output is on as long as an axis is at home.

I Indexer Step
Not yet implemented.

L Limit Error
The output goes low if a limit switch is hit during a normal move,or if both limits are hit during a Go Home move.

K,k At CL Limit (K specifies axis 1, k specifies axis 2)
The AT CL LIMIT output will become active when the motor current reaches the value defined by CL in the user's program.This output allows the Servo SmartDrive to indicate to a PLC when it has begun clamping,so that the user may synchronize the clamping operation with other processes in the application (for example drilling,cutting,or milling the clamped part).

M, m Move Complete (M specifies axis 1,m specifies axis 2)
The output goes high as soon as an axis move is started and goes low when a move is completed.

N aNalog
Only the OPTO I/O may be configured as analog outputs.To use an analog output module the position must be configured as an analog output.This tells the Smart Drive that the output is no longer a discrete output and assures that the output signal is sent properly.

Analog signals are set by assigning a value to reserved variables (AO9)-(AO16) corresponding to OPTPO positions 9 through 16.See Programming Your Application for more details on using analog I/O modules.

P Programmable
Unassigned outputs default to programmable and can be used in OT commands.

EDIT SETUP I/O **OPTOS**

Opto Configuration [OP]

Default:IIIIIIII

S Stall
The output goes low if the control detects a motor stall.

OPTO14: Output
IIIIOOOO ←↑↓→

T Torque Mode
Not yet implemented.

U Amplifier Unstable
Not yet implemented.

Z Z Direction
Not yet implemented.

The OPTOS menu allows you to configure each OPTO I/O position as either an input or an output.The hardware is protected against damage if you mistakenly configure an input as an output.

1. Use the ← and → keys to select an OPTO channel.The top line displays whether the channel is currently configured as an input or an output.
2. Once your cursor is on the desired channel,use ↑↓ to select INPUT or OUTPUT.

Your IDeal™ Programmable Smart Drive is compatible with almost any manufacturer's G4 or G5 digital OPTO modules (OPTO-22,Grayhill,Gordos,etc.).However, at the time of this printing only Grayhill's analog modules are compatible with our control.Other manufacturer's analog modules do not fit into a G4 footprint.

IDC stocks the following Opto modules,which may be specified when ordering an IDeal™ Programmable Smart Drive:

Order Code (p/n)	I/O Module Description	Opto-22 P/N	Grayhill P/N
A (PCB-1210)	10-32 VAC/VDC Input	G4IDC5	
B (PCB-1211)	TTL Input	G4IDC5K	
C (PCB-1212)	35-60V DC Input	G4IDC5G	
D (PCB-1213)	90-140 VAC Input	G4IAC5	
E (PCB-1214)	180-240 VAC Input	G4IAC5A	
F (PCB-1215)	5-60 VDC, 3 Amps Output	G4ODC5	
G (PCB-1216)	12-140 VAC, 3 Amps Output	G4OAC5	
H (PCB-1217)	Output 24-280 VAC, 3 Amps	G4OAC5A	
I (PCB-1218)	Input Test Switch	G4SWIN	
J (PCB-1219)	0-10 Volt Analog Input		73G-IV10
K (PCB-1220)	4-20 mA Analog Input		73G-II420
L (PCB-1221)	0-10 Volt Analog Output		73G-OV10
M (PCB-1222)	4-20 mA Analog Output		73G-OI420
N (PCB-1223)	J Type Thermocouple Input, 0 to 700˚C		73G-ITCJ
O (PCB-1319)	K Thermocouple Input, -100 to 924˚C		
P (PCB-1224)	RTD Thermocouple Input, 100 Ohm		73G-ITR100

More information on available OPTO modules is available from the Opto module manufacturer or your local distributor.

EDIT > SETUP > I/O > OUTSTS > **PWR-UP**

Configuring Ouput States on Power Up [OEP]

Default: OFF

```
- On PwrUp Output #1 -
←↓        OFF        ↑→
```

This option.sets the desired states of the outputs on power up.

1.Use ← and → keys to scroll throu outputs #1 – #8 and any OPTO positions configured as outputs.
2.Use the ↑ and ↓ keys to set the output state as OFF or ON and press ESC to register.

EDIT > SETUP > I/O > OUTSTS > **FAULT**

Configuring Ouput States on Fault [OEF]

Default: NO CHANGE

```
- On Fault Output #1 -
←↓    NO CHANGE    ↑→
```

This option.sets the desired states of the outputs on a fault.

1.Use ← and → keys to scroll throu outputs #1 – #8 and any OPTO positions configured as outputs.
2.Use the ↑ and ↓ keys to set the output state as OFF, ON or NO CHANGE and press ESC to register.

EDIT > SETUP > I/O > OUTSTS > **ST / K**

Configuring Ouput States on Stop/Kill [OES]

Default: NO CHANGE

```
- On ST / K Output #1 -
←↓   NO CHANGE   ↑→
```

This option.sets the desired states of the outputs on a Stop or Kill.

1.Use ← and → keys to scroll throu outputs #1 – #8 and any OPTO positions configured as outputs.
2.Use the ↑ and ↓ keys to set the output state as OFF, ON or NO CHANGE and press ESC to register.

EDIT > SETUP > I/O > **LIMITS**

Configuring End of Travel Switch Polarity [ET]

Default: NORM CLOSED

This option.allows configuration of the EOT switch polarity as NORM OPEN or NORM CLOSED to accommodate the use of either type of switch.

```
-- Axis One EOT Pol --
←↓  NORM CLOSED  ↑→
```

1.Use ← and → keys to select an axis.
2.Use the ↑ and ↓ keys to select NORM OPEN or NORM CLOSED and press ESC to register.

JOG Jogging Configuration

Your IDeal™ Programmable Smart Drive's keypad gives the programmer (and the machine operator if desired) a convenient way to jog the motor.

```
- - - ↑ JOG SETUP ↓ - - -
ACCEL   LO-VEL   HI-VEL
```

The parameters which control your jog operation are set using the JOG SETUP menu: Note:The Units used by the Jog parameters are configured from the SETUP-MECH menu.

EDIT SETUP JOG **ACCEL**

```
- - - ↑ JOG SETUP ↓ - - -
ENABLE
```

Jog Acceleration [JA]

```
- - Axis One Jog Accel
←        0.3 sec        →
```

Default:0.3 (Accel Units)

This option sets the acceleration and deceleration used during a jog move.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a new jog accel/decel value in the same units you selected in the SETUP/MECH/ACCEL menu.

EDIT SETUP JOG **LO-VEL**

Jog Low Velocity [JL]

```
- Axis One Jog Lo-Vel
←      0.5 in/sec      →
```

Default:0.5 (Velocity Units)

This option sets the low speed jog velocity used during a jog move.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a new low jog velocity value in the same units you selected in the SETUP/MECH/VEL menu.

EDIT SETUP JOG **HI-VEL**

Jog High Velocity [JH]

```
Axis One Jog Hi-Vel
←      2.0 in/sec      →
```

Default:2.0 (Velocity Units)

This option sets the high speed jog velocity used during a jog move.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a new high jog velocity value in the same units you selected in the SETUP/MECH/VEL menu.

EDIT SETUP JOG **ENABLE**

Jog Enable [JE]

```
- Axis One Jog Enable
←      ENABLED      →
```

Default:Enabled

This option enables or disables the jogging features of the control.When disabled,an error message is displayed when the jog buttons are pressed. Jogging functions are often disabled once a machine is installed to prevent operators from accessing them.

1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to enable and disable the function.

HOME Homing Configuration

```
-- ↑ HOME SETUP ↓ -
MODE  EDGE  SWITCH
```

```
- ↑ HOME SETUP ↓ --
OFFSET  FINAL
```

Your IDeal™ Programmable Smart Drive has a built-in homing function which combines the flexibility of a customized homing routine with the ease of use of calling a "canned" program. Also see the **GH** command in the IDeal Command Reference chapter for more details on homing.

EDIT > SETUP > HOME > **MODE**

Configuring Home Mode [HM]

```
- Axis One Home Mode -
←↓  SWITCH ONLY  ↑→
```

Default: SWITCH ONLY (Any OPEN LOOP encoder mode)
SWITCH THEN Z (Any CLOSED LOOP encoder mode)

The homing mode parameter establishes how a Go Home (GH) command will execute homing routines..

1.Use ← and → keys to select an axis..

2.Use the ↑ and ↓ keys to scroll through the list of homing modes and press ESC to select..

Homing Mode	Description
SWITCH ONLY	GH will only home to the appropriate edge of the home switch regardless of encoder mode.This is the only mode available without an encoder.
SWITCH THEN Z	GH will home to the switch,align to the edge and then slowly move until an encoder Z pulse is found.This mode requires an encoder.
Z CHANNEL ONLY	GH will slowly move until an encoder Z pulse is found.The state of the home switch is ignored.The magnitude of the GH velocity parameter is ignored.The sign of the GH velocity paramter determines the low speed direction.This mode requires an encoder.

EDIT SETUP HOME **EDGE**

Home Edge [HE]

- Axis One Home Edge -
← NEGATIVE →

Default:NEGATIVE

This option selects which side (positive or negative) of the home switch active region the Smart Drive must find before searching for the index channel of the encoder.

1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to select the active edge as the positive or negative side of the home switch.

EDIT SETUP HOME **SWITCH**

Home Switch [HS]

Axis One Home Switch-
← Norm Open →

Default:Normally Open

This option selects the type of switch used for the home input for each axis.A Normally Open switch connects to ground when activated.A Normally Closed switch is pulled high when activated.

1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to select the switch type.(NORM OPEN or NORM CLOSED)

EDIT SETUP HOME **OFFSET**

Home Offset [HO]

- Axis One Home Offset
← 0.0 rev →

Default:0.0 (Distance Units)

This option sets the home offset for each axis.After a successful homing move,the home position (the default home position is +0.0000) is set to the offset value.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a new home offset value in DIST units.

A home offset allows you to have separate systems with identical programs in them. All you have to change is the home offset value for each machine.It helps reduce start up time,since your home limit switch can now be almost anywhere.It also reduces the time necessary to get a system back up and running should your home switch ever get damaged or moved.

Example:Home Offset = 1.0000
When the control finds the home position,it sets the position counter to 1.0000 distance unit.The absolute zero position counter is now referenced 1 unit behind the mechanical home position.All absolute moves will be referenced from the absolute zero position.

EDIT SETUP HOME **FINAL**

Home Final Direction [HF]

```
- Axis One Final Dir. -
←      POSITIVE      →
```

Default:POSITIVE

Specify the final approach direction of your Go Home (GH) move with this option. This is the direction used to search for the encoder index mark (Z channel) after the appropriate home switch edge is found.

1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to select the final approach direction.

PROG Program-Run Configuration

```
- PROGRAM SETUP -
PWR-UP  SCAN  DELAY
```

Your IDeal™ Programmable Smart Drive is capable of running any pre-defined program on power-up, or whenever a valid BCD or binary program combination is read on inputs configured as BCD or Binary Program Select inputs (usually from a PLC or thumb-wheel switches).The conditions under which these programs execute are defined in PROGRAM SETUP.

EDIT SETUP PROG **PWR-UP**

Power Up Program [PU]

```
- - Power Up Program -
PROGRAM: 0
```

Range:n = 0 to 199 (0-400 w/ -30k option)
Default: 0

This option selects a power-up program.The selected program is executed (run) when your Smart Drive is powered up or reset.If a value of 0 is entered in this menu,or if the specified program does not exist,no program is run.
Use the numeric keys to enter a program number.

EDIT SETUP PROG **SCAN**

Scan Conditions [SN]

```
- - -Stop Scan After- - -
←↑ YYYYYYY ESC↓→
```

Default:YYYYYYY

The SCAN menu allows you to select which events will cause the control to stop scanning program-select configured inputs.It is used to enable or disable stop-scan events.If a given stop-scan event is enabled,the system will stop scanning the inputs for program numbers when that condition occurs.The Smart Drive must be reset via a Warm Boot input or by cycling power to start program scanning after an active Stop Scan event.This option has no effect if the inputs have not been configured as program select inputs (either BCD or Binary).

1. Each event is represented by one of seven Y/N positions on the bottom display line.Use ← and → keys to select a stop-scan condition.The selected event will be listed to the right of these 7 characters:ESCape,STOP, LIMIT+,LIMIT-,KILL, FAULT or INTerrupt.
2. Use ↑ and ↓ keys to enable (Y) or disable (N) the selected event.

EDIT SETUP PROG **DELAY**

Scan Delay [DY]

```
- - - Scan Debounce: - -
DELAY (ms): _
```

Default:100 ms

The DELAY time sets the amount of time the control requires the program select inputs (BCD or Binary) to remain stable before the control will recognize and run a program.The minimum time is 1 ms.If program select inputs are not stable for a time equal to or greater than the specified delay, the program will not be executed. Use numeric keys to enter a value in ms.

RS232 RS-232C Configuration

```
- - - RS232 SETUP - - -
ECHO  UNIT #
```

If you plan to use the serial communications port on your Smart Drive, you can use your keypad to turn the auto-echo on and off and set the unit's daisy chain address.The baud rate of 9600 is fixed on the Smart Drive.
Fixed RS-232 parameters:

- Baud rate:9600
- Data bits: 8
- Stop bits: 1
- Parity:none

EDIT SETUP PROG **ECHO**

Echo Enable [EC]

```
- - - - RS232 Echo - - - -
↑      ENABLED      ↓
```

Default:ENABLED

This option is used to enable or disable the RS-232C echo.If the ECHO is disabled, characters received by the control's serial port will not be re-transmitted.ECHO must be enabled in daisy chaining applications.
Use the ↑↓ keys to enable or disable the ECHO.

EDIT SETUP PROG **UNIT#**

Unit Number [UN]

```
 - - - Unit Address - - -
NUMBER: _
```

Default: 1
Range:(1-99) (Limited to 1-15 in firmware version 1.40 and earlier)

This option is used to set the unit address.Each unit in an RS-232C serial daisy chain of multiple units must have a unique Unit Address.Refer to the section on daisy chain operation in the RS-232 Operation chapter for more information on this type of application.
Use the numeric keys to enter the unit address.

MISC Miscellaneous Configuration Parameters

The following miscellaneous set-up parameters include auto-formatting of the keypad display, setting the deceleration rate used with a stop input,or the ESC key while an axis is moving,and disable/enable the TEST menus accessible from the RUN function key.

```
- -↑ MISC SETUP ↓- -
DISP  STOP-RATE  TEST
```

```
- -↑ MISC SETUP ↓- -
FAULT  ENABLE  PASWRD
```

EDIT > SETUP > MISC > **DISP**

DISP [DF]

```
< Quad #1 >Quad #2
  Quad #3   Quad #4
```

```
----Quad #1 Display ----
↑        INPUTS        ↓
```

Default: Quad #1: POS1.
Quad #2:POS2 (dual axis units)
BLANK (single axis units)
Quad #3:INPUTS
Quad #4:OUTPUTS

Display format allows teh user to customize the data displayed on the keypad run time screen.The run time screen has been divided into 4,10 character configurable quadrants.The DISP menu displays labels for teh 4 quadrants with carets (<>) denoting the selected quadrant.

1. Use the ←,→, ↑ and ↓ keys to move quadrant selection delimiters (<>)..
2. Press ENTER to edit quadrant.

Once a quadrant is selected,there are 16 possible data types that can be displayed in that quadrant.

Data Type	Quadrant Display
BLANK	No display
POS1	Axis #1 position
POS2	Axis #2 position
POS1+UNIT	Axis #1 position with axis units
POS2+UNIT	Axis #2 position with axis units
VEL1	Axis #1 commanded velocity
VEL2	Axis #2 commanded velocity
CUR1	Axis #1 current in Amps (B896X only)
CUR2	Axis #2 current in Amps (B896X only)
INPUTS	Discreet input status (0 off, 1 on)
OUTPUTS	Discreet output status (0 off, 1 on)
OPTOS	OPTO input and output status (0 off, 1 on) as configured
SA_STATUS1	Displays SA serial command response for axis #1
SA_STATUS2	Displays SA serial command response for axis #2
SS_STATUS	Displays SS serial command response
TEXT	Display user defined text in a quadrant

3. Use the ↑ and ↓ key to scroll through the data types and press ESC to register all data types except TEXT (see below).
4. In order to define a text field,scroll to the TEXT data type and then press the ALPHA key or a number key.A cursor will appear allowing up to 10 characters to be entered.Type the desired text,press the ENTER key and then press ESC to register.

EDIT SETUP MISC **STOP-RATE**

Stop Decel Rate [SR]

```
- Axis One Stop Decel
←      0.1 (rps²)      →
```

Default:100 rps^2 (units fixed at motor rps^2)

This option is used to set the deceleration rate for each axis whenever configurable stop input is activated,or when the ESC key is pressed while an axis is moving.

This is usually set to the fastest controllable deceleration rate possible with mechanics in your application.

1. Use the ←→ keys to select an axis.
2. Use the numeric keys to enter a stop deceleration.

EDIT SETUP MISC

TEST n/a

Option currently not implemented

Default:Enabled

Will be implemented in a future software release.
Enables and disables the RUN-TEST menu.Disable the test menu to prevent operators from accessing test functions such as I/O operation,trace mode and test moves.
Use the ↑↓ keys to change the setting.

EDIT > SETUP > MISC > **PASWRD**

CONFIGURING KEY PASSWORDS [PW]

--- PASSWORD SETUP ---
OPRATR ADMIN CLEAR

Default:None

In addition to the keypad dip switches,user definable passwords allow restricted access to the RUN, EDIT, COPY and DEL menus.

1. From the PASWORD SETUP menu,press F1 to enter an OPRATR password or press F2 to enter an ADMIN password.
2. Enter password using a maximum of 4 alpha-numeric characters only (0-9,A-Z and a-z only).See S696X,B896X and 96X manuals on how to enter alpha-numeric data on the keypad.

3. Press ENTER to register the password and ESC to exit.

Defined Passwords	Menu Accessibility
ORATR only	RUN, EDIT, COPY, DEL
ADMIN only	RUN, EDIT, COPY, DEL
OPRATR with ADMIN	RUN only (All RUN functions except TEST)
ADMIN with OPRATR	RUN, EDIT, COPY, DEL

Enter Password:
>

If no passwords are defined,there are no menu restrictions.If passwords are defined,pressing RUN, EDIT, COPY or DEL will display the password prompt.Entering the wrong password or pressing ESC at the password prompt will return the keypad to the standard run-time display. Selecting EDIT > SETUP > MISC > PASWRD > CLEAR will delete all passwords.

Password
USE LAST RESET

NOTE: Once a valid password is entered,the password prompt is replaced by the USE LAST, RESET prompt.Select USE LAST (F1) to use the last entered password or RESET (F3) to require the password to be reentered (i.e. require the next user to enter the password).This allows for subsequent use of the RUN, EDIT, COPY and DEL keys without requiring the password to be entered each time.

Tuning Your Servo System

Your B8961 or B8962 IDeal™ Programmable Smart Drive is a high-performance servo drive and motion controller in a single package. Your entire servo "system" includes the Smart Drive, motor and encoder. The encoder provides motor position and velocity feedback to the Smart Drive. You create a motion program using the IDeal programming language. The Smart Drive generates move trajectories and motor torque based on your high-level commands, which tell the motor where to go. It also takes care of closing the servo loops, which force the motor to follow the velocity and position profiles.

Since the B Series Smart Drive is a fully digital drive, motor data must be downloaded to it prior to operation. When an IDC MOTOR TYPE is selected from the EDIT/SETUP menu in the Keypad or Application Developer, motor data and tuning parameters are downloaded from the controller's non-volatile memory, to the drive's on-board memory. This data includes motor winding information, pole count, and encoder resolution. These files also contain default tuning parameters. The tuning parameters have been optimized for use with IDC motors and actuators. They work very well with inertia ratios up to about 15 to 1. Most applications, in fact over 80%, require **NO ADDITIONAL TUNING**.

The B Series Smart Drives also allow you to monitor the performance of the drive during machine integration. Each axis has an analog Monitor Port that can be configured to output position and velocity error, as well as commanded velocity, commanded torque, and actual torque, in real time, scaled to user defined units. The Monitor Ports, along with an oscilloscope, make a powerful diagnostic tool. The Monitor Port is a 12 bit analog signal, updated at 2 kHz.

IDC's DSP-based servo Smart Drives use an easy-to-understand digital servo algorithm, so tuning is straightforward. You also have the option of invoking two additional proprietary algorithms, Anti-Hunt and Inertial Anti-Windup which may further improve your system's performance.

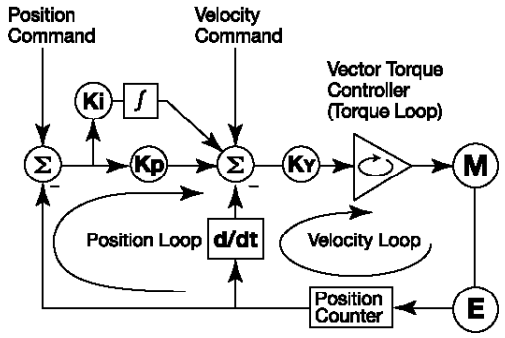
Your Smart Drive can be completely tuned with just the keypad and an oscilloscope. (Even the oscilloscope is optional for less demanding applications.) The keypad has two built in tuning functions. The first configures and scales the monitor output port. This allows you to view, on an oscilloscope, real time velocity and position error. No more "guessing" at the best tuning values. The second function outputs a torque, velocity, or position command square wave to the motor and mechanics. This extremely demanding move profile makes it easy to determine, and minimize, overshoot and response time.

Compensator

Before starting the tuning procedure,it is important to have a general idea of the control loop that you will be modifying.The structure of the B8000 control loop is very simple.The loops are nested similar to a traditional analog servo system. Because of this structure,the basic procedure for tuning a B Series Smart Drive is generally non-iterative.A simplified schematic of the proportional - integral - velocity loop (PIV) compensator is shown below.The compensator also contains velocity and acceleration feedforward terms that are not shown.

If tuning is needed for your application,it is a straightforward process.Please note that an oscilloscope is required to do any more than the roughest tuning.If an oscilloscope is not available,it is unlikely, except in very high inertia mismatch applications,that modifying the default tuning parameters will result in increased performance.

Tuning your system is a simple 3 step process.Each servo loop, velocity, position



and integral, must be tuned.Because of the implementation of the B8000 digital servo algorithm this tuning process is non-iterative.The loops are tuned from the inside out.The torque loop is motor/bus voltage dependent and is tuned at the factory.

1. Tune the velocity loop using a velocity toggle.
2. Tune the position loop using a position toggle.
3. Tune the integral loop using a position toggle (optional)

Details on how to do this are covered below and,in much greater detail,in the Servo Tuner manual.IDC also recommends accessing Servo Tuner's on-line help utility for the most up-to-date tuning procedures and hints.

RS-232C Users

- - - - - TUNING - - - - -		
AXIS1	AXIS2	RS232

Servo Tuner users will find that the Servo Tuner manual presents much of the same tuning information that is in this section.Please reference Chapter 7 for details on configuring a Smart Drive for RS-232 tuning,and the Servo Tuner manual included with this product for detailed tuning information and strategies.

Pressing the EDIT key followed by the TUNING function key produces a menu that looks like this:

Selecting RS-232 allows you to enable servo tuning via the Servo Tuner software. Servo Tuner also has an **Select/Axis1** or **Axis 2** pulldown menu to access Smart Drives without keypads.We provide this menu option to keypad users because there are additional tuning aids available with our Windows-compatible Servo Tuner software.

Key Pad Tuning

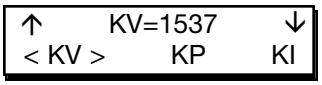
The Keypad tuning utilities were designed to mimic most of the capabilities of IDC's Servo Tuner software,but in a menu structure format.The default gain parameters generally provide adequate performance.If your application requires exceptional performance,whether it be low velocity ripple, fast settling times,or no steady state position error, you will may need to modify these standard gain parameters.

The keypad menu tree is shown below. Explanations on the use of the TORQUE, VELOCITY, and POSITION toggles can be found in the Servo Tuner manual.The same basic strategies presented there,apply to keypad based products as well.

Keypad Tuning Menu Tree

Axis Sub-Menus	Set-up Parameters	Description of Set-up Parameters
GAINS	KV	Velocity Gain
	KP	Position Gain
	KI	Integral Gain
	KFA	Feed Forward Acceleration Gain
	KFV	Feed Forward Velocity Gain
TOGGLE	TORQUE	Torque toggle (magnitude and period)
	VELOCITY	Velocity toggle (magnitude and period)
	POSITION	Position toggle (magnitude and period)
MONITOR	MODE	Configures monitor output function
	RANGE	Scales output function
NOHUNT	ON/OFF	Anti-hunt proprietary servo algorithm
WINDUP	ON/OFF	Anti-Inertial windup proprietary servo algorithm

GAINS Setting the Servo Gains





EDIT TUNING AXIS GAINS

There are five user accessible servo GAINS, available to tune your system.The gains are described below.

1. Use an F1,F2,or F3 to highlight the gain you wish to change.
2. Use the ↑↓ keys to switch to the other menu.
3. Use the numeric keys to Enter a new value for the gain that is highlighted.

If you need to tune your system,the fastest way to get top performance out of it is to tune it from "scratch." Set all your gains to zero,and follow the procedure outlined above,or in the Servo Tuner manual.

CAUTION: *This procedure does not work on load with steady state forces (such as gravity in a vertically oriented application). Call IDC for details on tuning these difficult loads.*

Toggle Tuning Stimulus

EDIT TUNING AXIS TOGGLE

```
POS   |0     |0
MODE  | STEP |  MSEC
```

1. Pick type of toggle;Press F1 to select a velocity, position,or torque toggle.
2. Magnitude of toggle.Press F2
   - Torque:% of max torque
   - Velocity:RPM
   - Position:steps (8000 steps/rev)
3. Period of toggle (in msec).Press F3
4. Start toggle.

Your motor and system will start moving when you press the RUN key.The toggle is stopped with the STOP (ESC) key.You can disable the amplifier by pressing the DEL key. Pressing DEL a 2nd time will enable the amplifier. If your system goes unstable because of too aggressive a toggle,or because of improper gain settings, disabling the amplifier is the only way to stop the out of control oscillations. Just pressing the ESC key and stopping the toggle won't stop an unstable system.These same toggle "hot"keys are active in the GAIN sub-menu so tuning changes can be made and evaluated while the system is actually moving.

CAUTION: *The toggle generates instantaneous torque, velocity, and position commands. The amplifier and motor will not be damaged, but your mechanics may be damaged by these very aggressive moves.*

Monitor Monitor Port

EDIT TUNING AXIS TOGGLE

1. Press MODE (F1)
2. Pick parameter to monitor with the ↑ and ↓ buttons.Choices include actual velocity, commanded velocity, position error, and actual torque.

```
-----EDIT MONITOR-----
MODE    RANGE
```

3. Press RANGE.Enter the range of values you want to monitor.The units depend on the parameter being monitored.

Connect your oscilloscope to Mon 1 (or Mon 2 for axis 2) and Mon Ref to view the selected parameter.This output is updated at 2kHz anytime the Smart Drive is powered up.Actual torque and velocity can be recorded while the program is running and the axis are moving.

Anti-Hunt (NOHUNT) and Inertial Anti-Windup (WINDUP) are two additional servo compensation algorithms which you can utilize with your Smart Drive.Details on these proprietary algorithms are provided in the Servo Tuner manual.

```
-- ↑ TUNING AXIS 1 ↓ -
NOHUNT   WINDUP
```

Programming Your Application

The purpose of this chapter is to provide information that will help programmers begin developing applications with the SmartDrive.There are also several practical examples that can be copied and modified.Other program examples are available from the Application De veloper disk set (see DEMOS.idc).

Smart Drive Programming Overview

Before beginning to develop a machine control program with a SmartDrive,the user must decide how the SmartDrive fits into the overall machine control hierarchy.There are three basic ways SmartDrives are used.They can be used in a standalone mode where the SmartDrive controls all the I/O and motion on a machine. They can also be used with a PLC,where the PLC runs the machine and just calls on the SmartDrive,via program select lines, for motion.Thirdly, they can be used in a ìhostedî mode with a PC.In this mode no programs are stored in the SmartDrive, the PC downloads a string of commands at execution time.

The information in this chapter applies to all three modes of machine control.The SmartDrive uses a sequential,interpretive,command processor.This means that commands in a program are executed one at a time,and that the action called for in one command must complete before the next command is processed.

Example:[move] VE4 DI10 OT01 GO OT10

In the program [move],the maximum move velocity is set to 4,the command incremental distance is set to 10,output 1 and output 2 are turned off and on simultaneously, axis one then moves 10 units.After axis one stops moving,output 1 is turned on and output 2 is turned off.The state changes on outputs 1 and 2 happen at the same time.

The programmer can control the flow of the program with WT (wait for an event or condition to occur),TD (wait for a pre-set amount of time to elapse),and IF (if a certain condition is true at this instant,then execute a block of commands) statements.External controllers such as PLC and computers can be coordinated via digital outputs and ASCII strings sent out the serial port.

Creating or Editing Programs with the Keypad

IDC's IDeal Command language is easy to remember and powerful.Command descriptions are also available on-line using the HELP key within the Application Developer program editor.

If you need help with basic keypad operation,please refer to Chapter 4 - Using the Keypad , and Chapter 5 - Configuring Your System .

Command Summary

The chart below lists all the IDeal commands that can be stored and executed as a part of a program.In Chapter 8, Programming with Serial Communication , there is more information on serial commands,such as Setup,Immediate Status, Supervisory Commands,and “Commands Not Available in Hosted Mode.”

Command	Description	Command	Description
AC	Acceleration	**IV**	Input Variable
BR	Break	**LP**	Loop
DA	Distance Absolute	**LU**	Loop Until
DC	Distance to a Change	**LW**	Loop While
DE	Deceleration	**MC**	Move Continuous
DI	Distance Incremental	**MS**	Message to Display
EA	Enable Amplifier	**ON**	On Command
EB	End of Block	**OT**	Output
EN	End of Routine	**RG**	Registration Move
FK	Function Key	**SP**	Set Position
GH	Go Home	**SQ**	Square Root
GI	Go Immediate	**ST**	Stop on Input
GO	Go (Start a Move)	**TD**	Time Delay
GP	Go Point	**VE**	Velocity
GS	Gosub	**WT**	Wait
GT	Go to	**“ “**	“Message to Serial Port”
IF	If Then		

Variables and Arithmetic

Variables

The Smart Drive will accept a variable in a command instead of a constant. Variables may be used in:

- Arithmetic
- Conditional Expressions
- Loop Counts
- Distance and velocity commands
- Set values
- Set command values or parameters
- Set analog signals
- Read analog or temperature input
- Display information such as position or velocity
- Any place that a number can be used,a variable can be used

Legal Variable Names

The Smart Drive allows you to create descriptive variable names,as opposed to V1, V2,etc. Variables can be up to 14 characters,but the first 10 characters must be unique.They can contain other printable ASCII characters,such as numbers, underscores, exclamation points, even spaces.Upper and lower case characters are supported within variable names,and these variable names are case sensitive.ASCII control characters such as LF and CR are not supported.All variables must be enclosed in parenthesis,(variable name). Parenthesis are not legal variable characters.

The standard software allows for up to 100 variables.All variables are stored as fixed point numbers.All variables are global.All standard variables are volatile, though non-volatile variables are available as well.

Built-In Variables

The following variable names are predefined in the control. They can be used throughout your programs in expressions,to set voltages,to test conditions,or even to display information on the keypad screen or some other external serial device.

Variable Name	Description	Type
(AI9) thru (AI16)	Analog Input 9 thru 16	Read Only
(AO9) thru (AO16)	Analog Output 9 thru 16	Write Only
(AROWREL)	Current status of any of the four arrow keys	Read Only
(CPOS1), (CPOS2)	Commanded Position of axis 1, 2	Read Only
(EPOS1), (EPOS2)	Encoder Position of axis 1, 2	Read Only
(POS1), (POS2)	Current Position of axis 1, 2	Read Only
(VEL1), (VEL2)	Commanded Velocity of axis 1, 2	Read Only
(EE1) thru (EE20)	Non-volatile, limited use, user system variables	Read/ Limited Write
(FKEY)	Value of Function Key pressed	Read Only
(LASTKEY)	Value of last Function key pressed	Read/Write
(TERM)	Sends variable out RS-232 port	Write Only
(1TW)	Scans inputs 1-4 for BCD Digit	Read Only
(2TW)	Scans inputs 1-8 for BCD Digit	Read Only
(TIME)	Elapsed Time (ms) since power up or since reset	Read Only
(CRCS)	Value of the EEPROM setup checksum	Read Only
(CRCP)	Value of the EEPROM program checksum	Read Only
(SA1), (SA2)	Integer value of the status of axis 1-2 (See RS232 command SA)	Read Only
(SD1), (SD2)	Integer value of the drive status of axis 1-2 (See RS232 command SD)	Read Only
(SS)	Integer value of the system status (See RS232 command SS)	Read/Write
(INT98CTRL)	Enables/disables (ARM INT98) trigger option - Refer to Ch. 5, Configuring Your Inputs, Input Descriptions, Input I.	Read/Write
(ARM INT98)	Enables/disables INT98 input if (INT98CTRL) is enabled	Read/Write

Examples of Built-In Variables

• (PIECES)=10	Assigns 10 to variable
• (SPEED)=(AI12)*(VEL SCALE)	Speed = analog input times a scalar
• MS21,"Enter Length"IV32,(LENGTH)	Prompts user and gets feed length
• VE(SPEED)	Sets velocity to value in variable
• MS1,(POS2)	Displays current position of axis 2 on keypad screen
• (TERM)=(POS1)	Sends the current position of axis 1 out the RS-232 port of the SmartDrive
• (TEMPERATURE)=(AI9)	Reads in temperature from analog input
• (AO15)=4012	Sets the analog output to 4012

Using the Built-In Variable (AROWREL)

(AROWREL) is a built-in Boolean read only variable which determines the status of any of the 4 arrow keys. When used in conjunction with (FKEY),the user can detect whether or not an arrow key is being held down. (AROWREL) will return one of the following values:

(AROWREL) = 0 One of the arrow keys is being held down.
(AROWREL) = 1 The arrow key has been released.

(AROWREL) will return key status for the 4 arrow keys only. If any other key is pressed,(AROWREL) will return zero regardless if the key is held down or not.The following is an example jog application using (AROWREL) and (FKEY):

```
[MAIN]          {Program #1}
FK12,13{Wait for a Left or Right arrow key}
GT(FKEY)        {Jump to arrow key program #12 or #13}

[LEFTARROW]     {Program #12}
MC+             {Enable MC mode}
AC.1            {Start MC move}
VE1             {Move in positive direction}
GO
LP
  IF(AROWREL)=1{Check status of arrow key}
    VE0 {Stop MC move on key release}
    GO
    GT1{Return to main program}
  EB
EB              {End loop block}

[RIGHTARROW] {Program #13}
MC+             {Enable MC mode}
AC.1            {Start MC move}
VE-1            {Move in negative direction}
GO
LP
  IF(AROWREL)=1{Check status of arrow key}
    VE0 {Stop MC move on key release}
    GO
    GT1{Return to main program}
  EB
EB              {End loop block}
```

Non-Volatile Variables

The non-volatile variables (EE1)-(EE20) are twenty user accessible variables that retain their values through power cycles, warm boots,and system resets.Standard user variables are reset at power down or reset.Each time one of these variables is changed (i.e.used on the left side of an equal (=) sign,the new value is written to and stored in the user non-volatile EEPROM.

CAUTION

Caution must be observed when using these variables. Since EE have a limited read/write lifetime (100,000 writes before failure), variable values that change frequently should not be stored as EE system variables. Examples include loop-count variables, and POS1 and POS 2 variables. The SmartDrive will allow only 1,000 EE writes between power cycles. This limit has been set to prevent a simple programming mistake or misunderstanding from permanently damaging the SmartDrive's non-volatile memory. When this write limit has been exceeded, all programs will stop running, an error message will be displayed, and the appropriate status bits will be set.

The EE system variables were originally developed for use in batch manufacturing applications where a number of variable setup parameters must be entered at the start of each part run.These same setup parameters can then be used through any number of power cycles,or machine resets.

Example:A program called [Set-up] is run at the start of each part run to initialize a number of variable part parameters.During production the program called [PARTS] is run.This program reads from the EE variables,but does not generate any writes to the EE,so the lifetime of the EE is not compromised.

```
[Set-up]
MS1,"Feed length?:"      Writes string beginning at character 1,top line
IV12,(LENGTH),1,15       Loads the part length into volatile user variable
                         (LENGTH)
MS1,"Feed Speed?:"       Writes string beginning at character 1,top line
IV12,(SPEED),.05, 5      Loads the speed into volatile user variable (SPEED)
(EE1)=(LENGTH)           Loads the length into non-volatile system variable (EE1)
(EE2)=(SPEED)            Loads the speed into non-volatile system variable (EE2)
EN
[PARTS]                  [PARTS] runs on power up,unless new parameters
                         are entered.
(LENGTH)=(EE1)           Load the part specific variable from the non-volatile
                         variables.
(SPEED)=(EE2)
LP(NUMBER)               Loop (NUMBER) of times
DI(LENGTH)               Move (LENGTH)
VE(SPEED)                at (SPEED) velocity
GO
OT1 TD.1 OT0             Toggle output to indicate part done
EB                       End the loop Block
```

Arithmetic Operands and Equations

The SmartDrive supports addition (+),subtraction (-), multiplication (*),and division (/).Expressions may only contain one operand.Complex equations require multiple statements. Variables and fixed point numbers may be mixed in arithmetic equations.All user arithmetic and variable storage uses 32-bit integer and fractional representation.

The + and - symbols have a dedicated button on the keypad.Pressing the button will toggle between the two.

The *,/,and = are accessed from the Alpha+0+... keystrokes.

Examples:
- (X)=(Y)*10
- (AO15)=(VOLTAGE)+(ERROR)

Do not enter:
- (X)=1+2-3 — This statement is not legal,because it has more than one operand.
- (Length)=(Total)*.03125 — The Smart Drive fixed point variable storage only supports 4 places to the right of the decimal place (32 bit storage of fractional decimal number).

Instead enter:
- (X)=1
  (X)=(X)+2 or (X)+=2
  (X)=(X)-3 or (X)-=3

and
- (Length)=(Total)*3.125 — Multiply by the significant figures.
- (Length)=(Length)*.01 — Then move the decimal place.
- **or** (Length)=(Total)/32 — The SmartDrive fixed point variable storage supports 4 characters to the right of the decimal place (32 bit storage of fractional decimal number).

Boolean Operators - & (And), | (Or)

The operators & and | will perform the respective bitwise Boolean functions on immediate or variable parameters.

An application example of the Boolean operators would be:isolating a specific bit from an SD (Tell Drive Status) response.Suppose you want to determine if axis #1 drive was enabled from a program.This corresponds to a bit #5 (10000) Binary, (16) integer in the SD response.The program segment would look as follows:

```
(DRIVE STAT)=(SD1)&16 IF(DRIVE STAT)=16 MS,1"Drive Enabled"EB
```

The 16 corresponds to an integer weight of bit #5 (10000) since you wish to "mask"out the enable bit.

Logical Operations on Expressions (&&,||)

Conditional commands (IF,WT, LU, LW) support logical operations of AND (&&) and OR (||).Two expressions may be logically AND'd or OR'd within one conditional command. For example:

(A)=5 (B)=2.5 IF(A)>2&&(B)=2.5 MS1, "True Statement" EB

In the above program,the message "True Statement" would appear since BOTH conditional statements are true,thus making the entire IF conditional true.

AND			OR		
Expression A	Expression B	A & B	Expression A	Expression B	A || B
False	False	False	False	False	False
False	True	False	False	True	True
True	False	False	True	False	True
True	True	True	True	True	True

Incrementing and Decrementing Variables (++, =+, --, -=)

There are four syntaxes supported by variables:++ (single increment),+= (value increment),-- (single decrement),-= (value decrement).These operators will initialize any uninitialized variable to zero before incrementing or decrementing it for the first time.

(Variable Name)++	Increments a variable value by 1
(Variable Name)+=n	Increments a variable value by n
(Variable Name)--	Decrements a variable value by 1
(Variable Name)-=n	Decrements a variable value by n

Expressions

The SmartDrive supports five conditional expressions:less than (<),equal to (=), greater than (>),less than or equal to (<=),and greater than or equal to (>=). The IF and WT commands can use these expressions to direct program flow or wait for an analog input to meet a condition.The > and < symbols are entered into the keypad editor with the ALPHA+↑ +↑ ....↑

Examples:

IF (X)>10 GS20 EB	If X is greater than 10 gosub to program #20
WT(AI12)<(MAX TEMP)	Wait for the temperature to go below the maximum before continuing command processing.

Summary of Operators, Functions and Expressions	
[ ]	Name Program
( )	Name Variable
&&	Logical AND
||	Logical OR
!	Logical NOT
!=	Not Equal
+	Add
-	Subtract
*	Multiply
/	Divide
=	Equal
>	Greater Than
<	Less Than
>=	Greater Than or Equal to
<=	Less Than or Equal to
&	Bitwise Boolean AND
|	Bitwise Boolean OR
++	Increment Variable
+=	Increment by n
--	Decrement Variable
-=	Decrement by n
<<	Shift Left
>>	Shift Right

Multi-axis Operation

IDC's IDeal command language is intuitive, yet powerful enough for difficult applications.The SmartDrives allow you to make multi-axis,synchronized,moves or multiple independent moves that start at the same time.

Simple GO Commands

The most intuitive and simple programs utilize the GO commands.Motion parameters for each axis are separated by commas. Motion parameters consist of the AC, DE,VE and either DI or DA command.These four parameters completely define the commanded motion profile.The last parameter used by an axis is stored in a table. The GO command uses the last DI or DA command processed to determine which axes move and how far they move. For example:

AC2,5	Set acceleration of both axes
VE10,3	Set velocity of both
DI9,-3.1	Set distance of each axis
GO	Start both together
DI9	Moves only axis 1 because the last D command
GO	only had a parameter for axis 1
DI,2	Moves only axis 2 because the last D command
GO	only had a parameter for axis 2

In the first move in the example,the GO command is used to start multiple axes at the same time.The move profiles for each axis are independent.While axes start at the same time,the moves are independent and will follow their respective accelerations and velocities,though the axes may not stop moving at the same time.

The Go Point (GP) command allows you to move two axes along a linear path that starts and stops at the same time.The control uses the most recent value in axis 1, the first parameter, to specify the AC,VE,and DE values along the path (see example 1 below).In most cases, your program will be easier to understand if you redefine these parameters as a single value for the path (see example 2 below).GP uses the last distance value for each axis to determine the end points of the linear move.The two examples below result in identical GP moves.

Example 1:	Example 2:
AC2,3	AC2
VE4,10	VE4
DE.5	DE.5
DI5,6	DI5,6
GP	GP

GO and GI commands support additional parameters providing more advanced features.The additional parameters allow you to:

1. Start each axis based on different conditions.
2. Start one axis with an input while another is moving.
3. Start one or more axis moving based on one input.
4. Start different axis based on different inputs.
5. Reduce the size of a program by using fewer “move”commands.

GO supports an extended syntax to allow these other move specifications.

GOi,i or GIi,i

i = 0-16,integers only

The extended syntax always explicitly defines which axes will move.Only those axes which have an active input defined by i will move,and the last move parameters will define the shape and length of the move profile.

An axis will start moving when the input defined by i turns on. You can start different axes at different times by using different integers (i) for each axis.

Setting i = 0 will explicitly define which axis will move when a GO command is processed.(See the examples below for more clarification.)

The last move parameters used to define a move profile are saved.This saves program space in applications where sections of code repeat the same moves.The DI or DA command is the only motion parameter that needs to be repeated.A stair-step pattern is one example of this.

Using different i's for each axis allows you to start one axis and wait for an input on another, or start both based on different inputs.See the examples below for more clarification.

Examples:

GO	Moves all the axes defined in the last DA or DI command
DI3 GO	Move only axis 1
DI3,4 GO	Move axes 1 and 2
DI2,2 GO,0	Move only axis 2, even if the last DA or DI command defined both axes
DA(Part 1),(Part 2)	
GO4,5	Move axis 1 once input 4 is activated, axis 2 once input 5 is activated
GO4,4	Move axis 1 and 2 once input 4 is activated

Typical Programming Examples

The following program examples will show how the IDeal command language can be used to perform simple tasks.There are more extensive and elaborate example programs in the DEMOS.idc file that came with your Application Developer disk set.

To aid your program documentation,comments can be placed in brackets {comment}.These comments are stripped out of the program as it is downloaded to help conserve memory in the control. Files should be saved BEFORE downloading for documentation purposes.

```
Example:  DI10,2 GO          {Moves to load position}
          DI15,15 GO         {Moves to unload position}
```

To create a Message and input a Variable

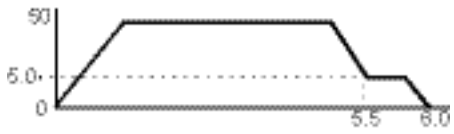
```
[GET PARTS]                  {Name the subroutine}
MS1,""                       Clears the display
MS1,"How many?:"             Writes string beginning at character 1,top line
IV12,(PIECES)                Waits at 12th character for the # of pieces
MS1,""                       Clears the display
MS1,"How long?:"             Writes string beginning at character 1,top line
IV12,(LENGTH)                Waits at 12th character for the length
LP(PIECES)                   Loops the number of pieces entered
DI(LENGTH)                   Moves the length entered
GO EB
```

Creating an Operator Menu *(see the FK command description for details)*

```
MS1,""                       Clears keypad screen
MS21,"PART1 PART2 PART3"     Writes a message above function keys
FK1,2,3                      Waits for a Function Key to be pressed
(FKEY)=(FKEY)+50             Add an offset to FKEY
GT(FKEY)                     Goto program #51,52,or 53.(50 + 1,2,or 3)
```

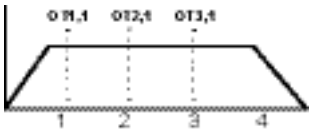
Fast In, Slow Feed Move *(Using the Distance to Change (DC) command)*

```
AC.05                        Set acceleration
DE.09                        Set deceleration
VE50                         Set first velocity
DA6                          Set total move distance
DC5.5                        Set point where you want to change speed
VE5                          Set second speed
GO                           Start the move profile
```



Turning On an Output on-the-fly

```
AC.05                    Set acceleration
VE10                     Set velocity
DA4                      Set total move distance
DC1                      Set point to turn on...
OT1                           Output 1
DC2                      Set point to turn on...
OT2,1                         Output 2
DC3                      Set point to turn on...
OT3,1                         Output 3
GO
```



To input a 4 Digit BCD number reading 2 Digits at a time

```
[GET 4 BCDS]                        Returns value of 4 digit BCD number
OT01                                Connect ground of first two BCD digits
(4 DIGIT BCD)=(2TW)*100             Make value of first two digits the MSB
OT10                                Connect ground of 2nd two BCD digits
(4 DIGIT BCD)=(4 DIGIT BCD)+(2TW)   Add value of 2nd two to 1st two * 100
```

Reading an Analog Input

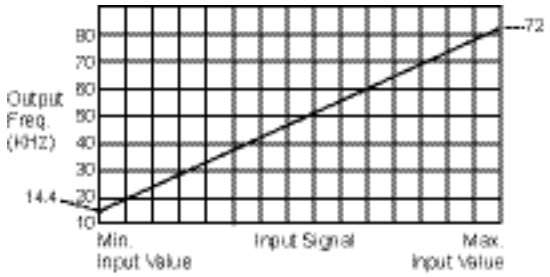
The value of the analog system variables (AI9-16) are scaled from 14,400 to 72,000 Hz.This value is actually a scaled frequency read by the SmartDrive.These input values are updated every 16 milliseconds .If your program needs to display this value in units such as VOLTS, you will need to scale the value to VOLTS in your program.The scale number you use will depend on which analog input you are using. For example:a "J" thermocouple uses a different factor than a "K" thermocouple.

Example:Using a 0-10 VDC analog input.0V=14,400;10V=72,000 or 5,760 Hz/volt.

```
(VOLT)=(AI12)              Read the value of analog input 12 into
                           variable volts
(VOLT)=(VOLT)-14400        Remove frequency offset
(VOLT)=(VOLT)*1.736        Scaling factor multiplied by 10,000
(VOLT=(VOLT)*.0001         Scaling back to volts
```

The variable (VOLT) is now in units of volts.If you are waiting for a condition to occur or doing a comparison,(see below) there is no need to go through the conversion process.

```
(TEMP)=(AI9)               Read in temperature from analog input
WT(AI12)<45000 GO          Wait for analog input 12 <45000 (<5.3 VDC
                           using the previous example) before moving
IF(AI12)<45000 GO EB       Go if analog input 12 < 45000
```



Configuring an Analog Output

The analog outputs of SmartDrives are 12-bit outputs.The value of the analog output is scaled to this 12-bit (0-4095) resolution.If you have a 4-20 mA output you will have an output resolution of .004 mA.

The output value is set with the system variable (AOn). n=9-16

The value of n corresponds to the OPTO position of the analog output.

To configure an OPTO position as an analog output:

1. Configure the position as an output using EDIT > SETUP > I/O > OPTOS.
2. Configure the output position as analog using EDIT > SETUP > I/O > OUTPUT.

The analog system output variable has a range of 0-4095 (12 bits).0 is the minimum output and 4095 is the maximum value of the analog module you are using.

Example:
4-20mA output module,0= 4mA,4095=20mA,

(AO15)=256 Sets the analog output to 5 mA

Programming Commands

This chapter defines,in alphabetical order, all of IDC's IDeal™ commands that can be used in a program.Please refer to Chapter 4, Using the Keypad, for more information on entering these commands with the keypad.

Some IDeal commands are supported only in serial communication mode.These commands are listed and defined in Chapter 8, Programming with Serial Communication , and they can also be found in the Summary of Commands immediately inside the back cover of this manual.

The commands in this chapter are defined according to the following example:

DI	**Distance Incremental**	**syntax - DI±r,±r**

Units: selected from the EDIT > SETUP > MECH menu (n/a if no units)
Range: unit scaling dependent (n/a if no range)
Default: If the command has a default,it will be listed here (n/a if there is no default)

DI specifies a move distance relative to the current position.Such moves are called incremental moves,as opposed to the absolute zero reference used in DA. Incremental moves are typically used in applications where there is no concern for origin,such as feed-to-length applications.Incremental moves are also often used inside a loop to shorten a program.Incremental and absolute moves may be mixed - the control always keeps track of the absolute position.

Example: Move 2 units in the + (positive) direction.
Move 1 more unit in the + direction.
Move 4 units in the - (negative) direction.

Program: AC.1 VE60 **DI2** GO **DI1** GO **DI-4** GO

Summary of IDeal™ Commands

Command	Description
AC	Acceleration
BR	Break
DA	Distance Absolute
DC	Distance to a Change
DE	Deceleration
DI	Distance Incremental
EA	Enable Amplifier
EB	End of Block
EN	End of Routine
FK	Function Key
GH	Go Home
GI	Go Immediate
GO	Go (Start a Move)
GP	Go Point
GS	Gosub
GT	Go To
IF	If Then
IV	Input Variable
LP	Loop
LU	Loop Until
LW	Loop While
MC	Move Continuous
MS	Message to Display
ON	On Command
OT	Output
“_”	“Message to Serial Port”
RG	Registration Move
SP	Set Position
SQ	Square Root
ST	Stop on Input
TD	Time Delay
VE	Velocity
WT	Wait

AC Acceleration — syntax - ACr,r

Units: sec,rps^2 or unit/s^2
(selected from the EDIT > SETUP > MECH menu)
Range: unit scaling dependent
Default: n/a

AC sets the acceleration and deceleration ramp on all velocity changes.The deceleration value (DE) will be the same as the acceleration value unless it is specifically set after the AC command.The value of DE must be reset every time AC is changed. Use only AC if you want a symmetrical move profile.Use DE if you want a different deceleration rate.Subsequent moves will use the last DE or AC value specified.

Examples:

AC2VE12 DA3 GO	Sets acceleration and deceleration to 2.
DE.5 VE12 DA6 GO	Accel stays at 2,decel changes to 0.5.
VE20 DA0 GO	Acceleration and deceleration remain at 2 and 0.5.
AC4 DA2 GO	Acceleration and deceleration become 4.
DE3 AC1 DI3 GO	DE reset to 1 by AC1 before the move is made.

BR Break — syntax - BR

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
Default: n/a

The Break command instantly "breaks"a loop block in which it is defined and continues program execution from the loop's terminating EB command.This allows for more complex loop conditioning than LU or LW commands.

Example:

```
(A)=0                          {Define variable A}
(B)=0                          {Define variable B}
LP                             {Define loop block}
  IF(A)>10                     {Check if A is greater than 10}
    IF2,0                      {Check if input #2 is off}
      BR                       {Break loop}
    EB
  EB
(A)++                          {Increment variable A by 1}
EB                             {BR command jumps here}
MS1,"A is greater than 10"     {Display message}
```

DA Distance Absolute — syntax - DA±r, ±r

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
Default: n/a

DA sets the next move position, referenced from absolute zero.The absolute zero position is established after a Go Home move (GH) and/or with the Set Position (SP) command.Absolute positioning is typically used in applications where you are moving to a number of known locations,or if your physical work area is restricted.

Incremental (DI) and absolute moves may be mixed;the control always keeps track of the absolute position.

Examples:

AC2 DE.5 VE12 DA3	GO Moves to absolute position 3 units.
DA3 GO DA3	GO Moves once to absolute position 3 units.

DC Distance to a Change — syntax - DC±r, ±r

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
Default: n/a

DC is used to define a complex, multiple velocity move profile,or to change an Output at a specific point during the move.It defines a position at which a change will occur,"on the fly," while the motor is still moving.At the specified position you can change the velocity or change the state of one or more outputs.

The DC command must follow a DA or DI command which specifies the total move distance.The DC distance is interpreted as an absolute position when used with DA and an incremental position when used with DI.When used with DI,the value of DC must be specified as a positive number.When multiple DC's are specified within an incremental move (DI),the incremental distance specified by the DC command is taken from the last DC command,not from the beginning of the move.See the incremental move examples below for more clarification.The standard software supports a maximum of 20 DC commands within a move profile.

Application Note: The DC command can only be used when the motor is moving at constant speed, i.e. not accelerating or decelerating. Issuing a DC command (or trigger position) before a previous DC has finished execution is invalid and can cause unpredictable results. For example,the following programmed move profile is an incorrect use of the DC command: AC1 VE5 DA20 DC1.75 VE7.5 GO

Since the initial acceleration ramp requires 2.5 units of distance to reach velocity $(S=\frac{1}{2}V)$, the DC1.75 is an invalid trigger position and will be ignored.

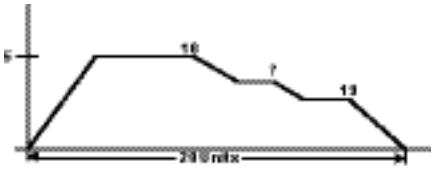
The following formula ensures the use of valid DC trigger positions:

$$DC_n - DC_{n-1} - \frac{|V_{n-1} - V_n|^2}{2} \geq 0$$

Where n is the current DC command distance (n=19 in this example), n-1 is the previous DC command distance (e.g.10), V is the velocity, and t is acceleration time (for the first DC specified in a move profile, n-1 would correspond to the beginning of the move).

In the following move profile, AC units = seconds, VE units = RPS (see illustration):

AC1.6 DE0.8 VE5 DA20 DC10 AC2.5 VE3 DC? VE2.5 GO



Using the DC formula and solving for DCn, $DC_n = \frac{(|5-3|)(2.5)}{2} + 10 = 12.5$ therefore, the "?" must be greater than or equal to a position of 12.5 distance units. In addition to verifying the DC trigger position, it must also be verified that the DC? VE2.5 segment can be completed before the beginning of the move deceleration. Suppose the "?" was chosen to be 13.35 (a valid trigger position), use the beginning of the decel ramp as the DC_n in the DC formula. A deceleration from 2.5 to 0 requires 1 distance unit in 0.8 seconds $(S = \frac{1}{2}Vt)$ therefore, $19 - 13.35 - \frac{(|0-2.5|)(0.8)}{2} = 5.65$ Since the result is positive, the DC13.35 VE2.5 is a valid segment.

Examples of DC move profiles:

AC.05 DE.05 VE10 DA4 DC1 OT100 DC2 OT010 DC3 OT001 GO
While moving to an absolute position of 4 units turn on output 1 at 1 unit, output 2 at 2 units and output 3 at 3 units.

AC.05 DE.09 VE30 DA6 DC3VE15 GO
Move to absolute position 6 units with a starting speed of 30. At 3 units, reduce speed to 15 (change-on-fly) and complete move.

AC1 DE.5 VE20 DI-8 DC1 OT10 DC3 OT01 GO
Move an incremental distance of negative 8 units. After 1 unit, turn on output 1, and after 3 additional units of motion, turn off output 1 and turn on output 2.

AC.05 DE.15 VE50 DI15 DC5VE10 DC5VE5 GO
At a starting speed of 50, begin moving an incremental distance of 15 units. After 5 units, ramp down to 10 speed. After an additional 5 distance units, ramp down to 5 speed and continue until the final position is reached.

AC.1,.1 VE10,20 DI10,10 DC5,5VE20,10 GO
Move an incremental distance of 10 units on axis 1 and 2. After 5 units on axis 1, ramp up to a speed 20; after 5 units on axis 2, ramp down to speed 10. Note that these velocity changes both take place after 5 distance units, but they do not take place at the same time.

DE Deceleration — syntax - DEr,r

Units: sec,rps^2 or unit/s^2
Range: unit scaling dependent
Default: n/a

Sets the deceleration ramp on all negative velocity changes. The deceleration value will be the same as the acceleration value unless a deceleration is specified.The value set will be used on subsequent moves unless it is re-specified by either an acceleration (AC) or deceleration (DE) command.

Examples:

AC2 VE12 DA3 GO	Sets acceleration and deceleration to 2.
DE.5VE12 DA6 GO	Accel stays at 2 and decel changes to 0.5.
VE20 DA0 GO	Acceleration and deceleration remain at 2 and 0.5.
AC4 DA2 GO	Both acceleration and deceleration become 4.
DE3 AC1 DI3 GO	AC1 sets both the accel and decel to 1.

DI Distance Incremental — syntax - DI±r, ±r

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
Default: n/a

DI specifies a move distance relative to the current position.Such moves are called incremental moves,as opposed to the absolute zero reference used in DA. Incremental moves are typically used in applications where there is no concern for origin,such as feed-to-length applications.Incremental moves are also often used inside a loop to shorten a program.Incremental and absolute moves may be mixed - the control always keeps track of the absolute position.

Example: AC.1 VE60 DI2 GO DI1 GO DI-4 GO
Move 2 units in the + direction.Move 1 more unit in the positive direction. Move 4 units in the negative direction.The final absolute position is -1.0000.

EA Enable/Disable Amplifier — syntax - EAi,i

Units: n/a
Range: 0 (disable),1 (enable),2 (standby)
Default: n/a

EA sets the state of the amplifier enable signal.The polarity can be changed in EDIT > SETUP > MISC > ENABLE.

Example: EA0 — Disables the amplifiers on axis one and two.

EB End of Block — syntax - EB

Units: n/a
Range: n/a
Default: n/a

The EB command designates the **E**nd of a **B**lock of loop or IF commands.Every LP, LW, LU, and IF statement must have an EB associated with it.

Examples:

LP2 DI3 GO EB	Performs the move twice
IF1,1 DI5 GO DI10 GO EB GH3	If input 1 is On,make 2 moves before homing. If input 1 is Off, jump to the GH command.

EN End of Routine — syntax - EN

Units: n/a
Range: n/a
Default: n/a

EN marks the end of a program or subroutine.It is optional at the end of a program.If EN marks the end of a subroutine,command execution continues from the command following the gosub (GS) command which called the subroutine.If the routine was not called from another program,the EN command simply stops command execution.The control continues to monitor the program select inputs (if defined).

The EN command can be used anywhere in a program to stop command execution.

Example:

IF2,1 EN EB DI2 GO	If input #2 is on,stop the program,or return to the calling program.If not,move 2 units.

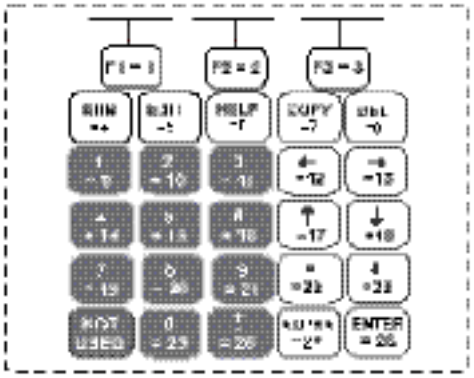
FK Function Key — syntax - FKi,i,...i

Units: n/a
Range: i = 1-28
Default: n/a

Note:24,the ESC key, cannot be assigned since it stops a program.
The FK command allows you to redefine a keypad key function within your program.The FK command pauses processing until the buttons you have "armed"are pressed.The number of the armed button is assigned to the system variable, (FKEY).You can then manipulate or directly use this variable to branch to other routines or make other decisions.FK allows the programmer to redefine the keypad function keys as operator menu selection buttons.You can even write your program with menus that look and feel like our setup menus.

Example: FK1,2,3,4
GS (FKEY)
Pauses command execution until F1,F2,F3,or RUN is pressed on the keypad.(FKEY) is assigned a value of 1-4.
Subroutine 1-4 is called with the GS (gosub) command.

See the illustration below for the value of (FKEY) returned for each key:



The following example shows how to use the keypad function keys as an operator interface:

1. Write a menu message (MS) on the keypad display above the corresponding function keys.
2. Use the FK command to pause command processing until the operator selects a valid function key. Only keys explicitly defined in the FK statement are considered valid.
3. Gosub to the appropriate program.

Example of a 3-screen menu program:

```
Program 20:
[SCREEN 1]                          Name the main program
MS1,""                              Clears keypad screen
MS3,"Select a Part"                 Writes a Message
MS21,"Part A  Part B  Part C"       Writes a message above function keys
FK1,2,3,17,18                       Wait for selected key press
GT(FKEY)                            Jumps to prog # 1,#2,or #3 if F1,F2,
                                    or F3 is pressed
                                    Jumps to prog #17,or #18 if the up or
                                    down arrow keys are pressed.
EN                                  End of Routine

Program 18:
[SCREEN 2]
MS21,"Part D  Part E  Part F"       Writes a message above  F1,F2,F3.
FK1,2,3,17,18                       Wait for selected key press
IF(FKEY)=17 GT[SCREEN 1] EB   If Up arrow goto screen 1
IF(FKEY)=18 GT[SCREEN 3] EB   If Down arrow goto screen 3
(FKEY)=(FKEY)+3                     Add offset to FKEY variable to goto
                                    correct part subroutine.
GT(FKEY)                            Jumps to part D, E,F in program
                                    #4,5,or  6
EN                                  End of Routine
```

```
Program 17:
[SCREEN 3]
MS21,"Part G  Part H  Part J"          Writes a message above function keys.
FK1,2,3,17,18                          Wait for selected key press
IF(FKEY)=17 GT[SCREEN 2] EB   If Up arrow goto screen 2
IF(FKEY)=18 GT[SCREEN 1] EB   If Down arrow goto screen 1
(FKEY)=(FKEY)+6                        Add offset to FKEY variable to goto
                                       correct part subroutine
GT(FKEY)                               Jumps to part G, H,J in program
                                       #7,8 or 9
EN                                     End of Routine
```

The programs to make Parts A,B,C, D, etc.are in program numbers 1-9. To continuously cycle through put a GT[SCREEN 1] at the end of each part program.

GH Go Home — syntax - GH±r,±r

Units: velocity units selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
Direction: positive (+) direction established in EDIT > SETUP > MOTOR menu
Default: n/a

The GH command initiates a homing routine (seeks the home switch) to establish a home reference position.When it reaches home,the position counter is set to zero or to the Home Offset value selected in the EDIT > SETUP > HOME menu.

The motor will move at the GH velocity (n) and direction (±) specified until it either finds a home limit switch or determines that it cannot find one between the two end-of-travel limit switches.The Go Home move uses the last acceleration and deceleration specified.

The exact homing routine used,and the ultimate end position of your system's home reference,depends upon the values of your EDIT > SETUP > HOME parameters (edge,level, final approach direction,and offset,) and whether or not you have specified open or closed loop moves in the EDIT > SETUP > ENCODER menu.

The control will reverse direction when the first End of Travel limit switch is encountered while searching for a Home switch.If the second End of Travel switch is encountered,the unit will abort the Go Home move and generate a fault.

Assuming the presence of an operational home switch,the control will ultimately seek a home position according to the home setup parameters you specified (edge, level, final approach direction,and offset).

Closed loop systems will normally home with more accuracy than open loop systems because encoders come with a Z marker pulse (1/8000 of a revolution on our B Series.) In a typical Go Home routine,the control will first sense the edge of the switch defined in the Go Home SETUP menu.It will then decelerate the motor to a stop at the last defined deceleration rate.The final homing motion will now be determined by the Go Home options selected in the SETUP menu.

The final homing direction dictates the direction from which the final approach to the switch is made.The edge selected will determine from which side of the home switch this final approach will be based.In a "closed loop"mode Go Home routine, the control will additionally slow to a creep,accelerate and stop when it sees the encoder's "Z"Marker Pulse after seeing the reference edge of the switch.If a mark-

er pulse is not seen within one motor revolution after the reference edge of the switch is seen,the final homing routine will be aborted.

Note: Homing Mode directly affects or reconfigures the function of the GH command (see "Configuring Your Home Parameters"in Chapter 5).

Examples:

AC.5 DE.5 GH-20	Go Home in the negative direction at a speed of 20
AC.5 DE.5 GH20	Axis one Go Home in the positive direction at a speed of 20. Axis two Go Home in the positive direction at a speed of 18.

Application Note from the IDC Applications Engineering Group - Regarding Simultaneous 2-Axis Homing Routines (e.g. GH5,5)

When the move is executed,the control looks at the Home 1 and Home 2 inputs. This homing routing has three phases:

1. Going Home (initial move toward Home switch).
2. Shuttle Move (reverse and search for first switch edge).
3. Low Speed Home (move to final switch edge - positive or negative).

Both axes will not complete their homing routines independently.When a Home switch is triggered on axis 1, for example,axis 1 stops and waits for axis 2's Home switch to be triggered.Basically, the homing routine will not move to the next phase (Shuttle Move or Low Speed Move) until both axes have completed the current phase.

If type "H"and "h"outputs (At Home 1 and At Home 2) are being used,they will both turn on when the entire homing routine has been completed,not individually.

GI — Go Immediate — syntax - GI or GIi

Units: n/a
Range: n/a
Default: n/a

The GI command begins a defined move profile in the same manner as the GO command.Unlike the GO command,where program execution waits until all defined moves have terminated,GI allows program execution to continue once the move has begun.This allows for other program defined processes to take place while an axis is moving,such as independent multi-axis moves, OT commands,and conditional IF blocks.One axis is not required to wait for another axis to finish a move before beginning its own move.

Following is an example of a program using the GI command:

VE1 DI20 GI MS1, "Axis #1 is moving" TD2

In this example,once the DI20 move begins,program execution immediately displays the "Axis #1 is moving"message for 2 seconds.Once the TD2 command has executed,the program will terminate;however, axis #1 will continue to move until the DI20 distance is reached.A Stop,Kill,or press of the ESC key will halt a GI based move either inside or outside program execution.

The GI command can cause program execution and moves to be asynchronous.In order to re-synchronize the end of a GI move with program execution,use the Wait (WT) command and its new syntax,i.e.WT#1 will wait for only axis #1.

If a program error occurs during a GI move,the move will stop at the Stop Decel Rate.

USING THE GI COMMAND - EXAMPLES

The following examples are provided to help further explain the use of the command:

A. If a GI move is in progress and an additional move is commanded on the same axis,the additional move will not begin until the GI move has completed. For example:

VE1 DA100 GI OT1,1 DA0 GI IF1,1 MS1,"All moves done" TD5 EB

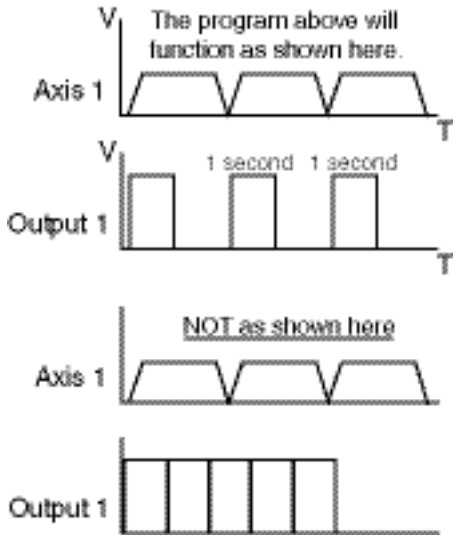
In this program,one may expect to see the message "All moves done"immediately after the DA100 move begins.In reality, the program will wait at the DA0 GI until the DA100 move has completed before continuing.More simply stated,a move cannot be commanded to begin on an axis that is already moving.

B. For multi-axis systems,GI allows moves within a move. For example:

VE1 DA100 GI LP5 VE,10 DI,5 GO EB

C. Since GI allows program execution to continue,there can be programming issues when using GI. For example,in the following program fragment:

LP VE2 DI10 GI OT1 TD1 OT0 EB

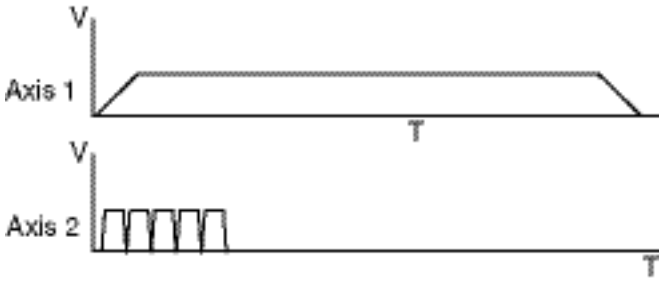


After the first pass through,the loop command (LP) will wait at the GI command since subsequent GI moves must wait for the present move to finish.

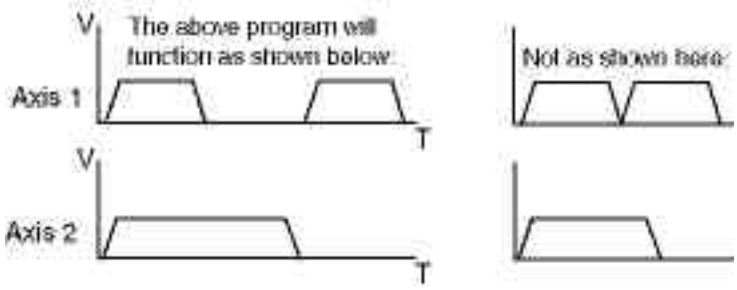
D. Another issue is programming conditional GI moves within IF blocks, for example:

LP IF1,1 VE5 DI20 GI OTXX100 EB IF2,1 MC,+ VE,10 GO EB IF3,1 VE,0 GO EB EB

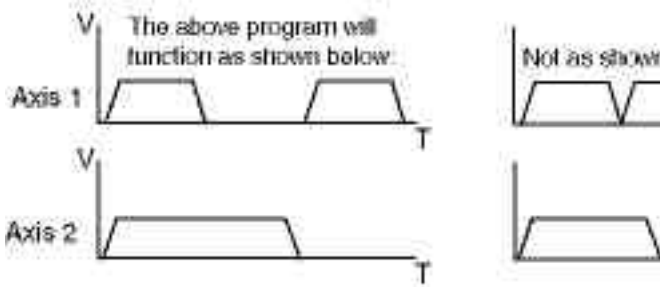
In this program,when input #1 is asserted,the GI move will be commanded,however, since the program will continue executing,input #1 may still be asserted on the next loop iteration.This will cause a second GI move to be commanded which may be undesirable.To avoid this situation,the addition of a WT1,0 after the IF1,1 will ensure that input #1 is only seen once since it has to be deactivated to allow the program to continue.



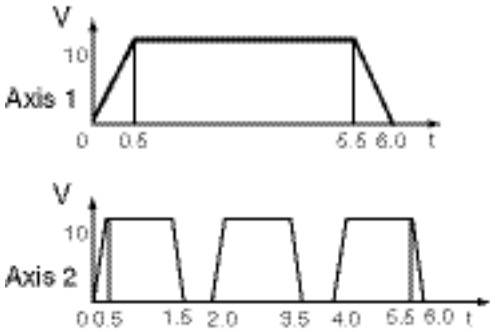
E. **DI1 GI DI,5 GO DI1 GI**



F. **DI10 GI DI,2 GO DI,2 GO DI10 GI**



```
G.  (VAR1)=10                 {Initialize Variable}
    LP                        {Beginning of Loop Block}
        DI55,10               {Define Two Axis Move}
        VE10,10
        AC.5,.5
        GI                    {Start Go Immediate Move Both Axes}
        MS1,""                {Clear Screen}
        MS1,(AI9)             {Write Analog Input 9 to the Screen While
                               Moving}
        WT,#2                 {Wait for Axis 2 to Stop Moving}
        TD.5                  {Time Delay of 0.5 seconds}
        DI,10                 {Define Axis 2 Move of 10 units}
        GI                    {Start Axis 2 GO Immediate Move}
        IF8,1                 {If Input 8 is on}
        OT10,1                {Turn on Output 10}
        TD.1                  {Time Delay of 0.1 seconds}
        OT10,0                {Turn off Output 10}
        EB                    {End of If Block}
        WT,#2                 {Wait for Axis 2 to Stop Moving}
        TD.5                  {Time Delay of 0.5 seconds}
        DI,10
        GI                    {Start Axis 2 Go Immediate Move}
        (DIST)=(VAR1)*1000    {Do Variable Math while Moving}
        (TERM)=(DIST)         {Send Value of DIST Variable out of Serial Port}
        WT,#2                 {Wait for Axis 2 to Stop moving}
        OT2,1                 {Turn On Output 2}
        WT#1                  {Wait for Axis 1 to Stop Moving}
        OT1,1                 {Turn On Output 1}
        DA0,0
        GO                    {Move Both Axes Back to Starting Position}
    EB                        {End of Loop Block,Restart Loop}
    EN                        {End of Program}
```



GO — Go (Start a Move) — syntax - GO or GOi,i

Units: n/a
Range: i = 1-16
Default: n/a

GO executes a move profile defined by some combination of AC,VE,DE,DI, DA, DC,or MC commands. Actual motion of a new profile will occur after a short calculation of the motion trajectory.

GOi pre-calculates the move and waits for Input number "i"to activate before executing.This variation is sometimes useful for applications needing very short, repeatable move calculation delays.It is more often used simply to shorten code, since it functions like the combination of Wait on Input and Go (WTi GO) yet it pre-calculates the move.Like other commands using I/O, GOi does not restrict you from using an input even if it has been configured for some predefined function.

Example:

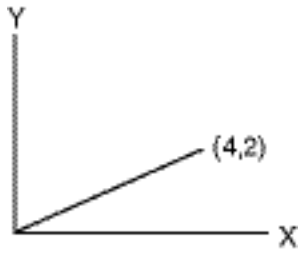
AC.05 DE.05 VE50 DI5 GO	GO initiates calculation of a move profile using buffered parameters (.05 unit Accel and Decel Ramp,speed 50,5 unit incremental move) and then executes it.
AC.05 DE.05 VE50 DI5 GO2	When input 2 is activated,immediate execution of the motion calculation already in the buffer is performed.

NOTE: If you have a multi-axis control refer to the section on multi-axis moves in Programming Your Application. It contains important information on multi-axis syntax and synchronization.

GP — Go Point (Linear Interpolation) — syntax - GP or GPi

Units: n/a
Range: n/a
Default: n/a

The GP command allows the SmartDrive to execute a linear interpolated move. An example of the GP move is as follows:



The units and values for path velocity, acceleration and deceleration in GP moves are specified by the parameters traditionally defined for an axis #1-based move.As in regular moves, velocity, acceleration,and deceleration only need to be defined once.Each sequential GP move thereafter will use these values until new values are specified.The endpoint of the move is specified by a two-axis DA or DI command

which corresponds to the appropriate X and Y coordinates. The following program would execute the move in the illustration above: VE2 AC.1 DA4,2 GP

The path velocity is 2 user units/sec,path acceleration is 0.1 sec.,and the X,Y position would be (4,2).

The GPi syntax is identical to the GOi command.GPi pre-calculates the move and waits for Input number "i"to activate before executing. For example,GP4 would pre-calculate the move and wait for input 4 to be active before executing.

Although both axes move during a GP defined move,all GP parameters refer to the path movement rather than to the individual axis movements.

NOTES ON GP:

GP will work with any velocity and acceleration unit.

The largest possible GP moves are restricted to $X^2 + Y^2 <= (2^{31} - 1)^2$ in units of steps. For example,the longest simultaneous X,Y point move is:

Steps:	DA1518500249,1518500249
User units with resolution 8000:	DA189,812.5311,189,812.5311
User units with resolution 25000:	DA60,740.0099,60,740.0099

Commanding moves larger than $X^2 + Y^2 <= (2^{31} - 1)^2$ will produce unpredictable results.

The DC command does not recognize an interpolated move as a 'single'move and will treat the axes independently. Therefore,using a DC during a GP move will cause unpredictable results unless the user has calculated the necessary values to preserve the vector move.

GS Gosub syntax - GSi and GS[name]

Units: n/a
Range: i = 1-199 (1-400 with 30K memory option),[name] = any legal program name
Default: n/a

Jumps to program number or name and returns to the calling program when command processing reaches the EN command in the sub-routine. After the return, execution continues at the command immediately following the GS.Subroutines may be nested in the standar d firmware up to 16 levels deep.A Goto (GT) will clear the subroutine stack,preventing future Gosubs from overflowing the stack or returning to the wrong location.

Example:

DI10 GS[Part A] GO	Run program "Part A", return and make a 10 unit incremental move.

GT — Go to Program — syntax - GTi or GT[name]

Units: n/a
Range: i = 1-199 (1-400 with 30K memory option),[name] = any legal program name
Default: n/a

GT branches to the program number or name specified.All subsequent commands in the calling program are ignored. Nested loops and subroutines calls are cleared by a GT command.

Example:

IF10 GT[PART A] EB	IF input 1 is on and input 2 is off, jump to program "Part A"
IF01 GT20 GT30 EB EN	IF input 1 is off and input 2 is on, run program 20.Program 30 will never run.Use the GS command if you want to return to this program and goto program 30.

IF — If — syntax - IF (See Below)

Units: n/a
Range: see below
Default: n/a

Syntax(s): IFi,xx...
IFxx... (assumes first input is input 1)
IF (mathematical expression) or expressions (2)
Range: i = starting input number, 1-8
x = 0;input high
x = 1;input low (grounded)
x = anything else (input level don't care)
expression = any valid expression (defined in math and variables section).

Allows the conditional execution of a block of commands based on the evaluation of an expression or input state.If the expression or input state is TRUE,the commands between the IF and the EB are executed.If FALSE, execution continues with the command following the EB.An IF statement should not be confused with a WT statement.An IF statement evaluates,true or false,based on the conditions that the SmartDrive sees at the instant the command is processed. A WT statement pauses command processing until the condition is true.

Note: An End of Block (EB) command must be used with every IF command.

IF blocks can be nested up to 16 levels deep.

To increase flexibility (primarily with programmable logic controllers or PLCs) the IF command allows you to use configured inputs in the command.To help prevent this added flexibility from causing programming confusion, you can specify any character as an input (x).This allows you to self-document your IF statements. For example,assume you configured input #3 as a "JOG SPEED"input.Programming such as "IF01J10"can help remind you that you are already using input #3 as "JOG SPEED".

Example:

IF14,1 GO EB	If input 14 equals 1 Go
IF12,010 GO EB	If inputs 12-14 equal 010 Go
IF110 GO OT3,1 EB	If inputs 1-3 equal 110 Go and turn on Output #3
IF(A19)<5500 OT11 GO EB	If analog input 9 is less than 5500,turn on output 1 and 2,then GO
IF(TEMP)>50 OT1 EB	If temperature variable >50 turn on Output 1
IF(PARTS)=25 GS20 EB	If PARTS variable = 25 Gosub to Program 20

IV Input Variable — syntax - IVi,(variable),*min*,*max*

Units: n/a
Range: i - 1-40 display position in characters
variable = any legal variable name
min = the minimum range value (optional)
max = the maximum range value (optional)
Default: n/a

This command allows an operator to input variable information under program control.It is typically used with the message command,MS,to prompt for operator input of the variable specified in IV.The cursor is placed on the display at character position "i".The program waits until a number is entered before continuing execution.The command will not allow you to type past the end of either line on the display.Variables will store 4 digits to the right of the decimal place.

When minimum and/or maximum range values are specified,the IV command will not accept inputs from outside this range. When a value outside the range is entered,one of the following messages is displayed on the keypad:

- "Input below minimum,Press ESC to resume"
- "Input above maximum,Press ESC to resume"

These variables can then be used in a math equation,conditional expression,or to set any command parameters (Example: DA,DC,VE, AC,LP, IF,TD, etc.).A variable can be used anywhere in a program where a real number or integer could be used.

Due to the nature of converting decimal numbers to binary and back,care must be taken in performing math on variables used in LP statements.LP will truncate the non-integer portion of the variable. For example:(COUNT)=25*.2 LP(COUNT) will only loop 4 times because (COUNT)=4.9999.A small offset can be added to variables used in LP statements to avoid this problem.(COUNT)=(COUNT)+.1 will guarantee that (COUNT) will be greater than 5,so the program will loop 5 times.

Example:

MS1,""	Clears the Display
MS1,"How many?:"	Writes string beginning at character 1,top line
IV12,(PIECES),1,15	Waits at 12th character for the # of pieces in the range 1-15.
MS1,""	Clears the Display
MS1,"How long?:"	Writes string beginning at character 1,top line
IV12,(LENGTH)	Waits at 12th character for the # of pieces.
LP(PIECES)	Loops the number of pieces entered
DI(LENGTH)	Defines the desired move length/distance.
GO	Moves the length commanded
EB	Ends the loop.

LP Loop syntax - LPi

Units: n/a
Range: n/a
Default: 0

LP will cause all commands between LP and EB to be repeated "i"times. If LP is followed by a 0 or no number, the loop will repeat continuously.

Note: An End of Block (EB) command must be used with every LP command.

The standar d software allows up to 16 nested loops (one inside the other). Each LP command must have a corresponding EB command to end the block (loop). A GT command within a loop will terminate the loop, clear the loop stack, and jump to a new program.

Example: AC.09 DE.09 LP3 VE30 DI1 GO EB VE7 DI-3 GO EN

The motor will perform an incremental 1 unit move at speed 30 three times and then a 3 unit move at speed 7 in the other direction.

LU Loop Until Condition True syntax - LU (See Below)

Units: n/a
Range: n/a
Default: n/a

The Loop Until (LU) command defines a loop block in which loop iterations are based on a conditional result.The syntax for LU, which is identical to the IF command,is as follows:

Syntax: LUi,xx…
LUxx…
LU(Mathematical expression) or expressions (2)
where,
i = starting input number, 1-8 (SmartStep),1-16 (SmartDrive)
x = 0,(Input Off)
x = 1 (Input On)
x = anything else (Input level Don't Care)
Mathematical expression = Any valid conditional or logical expression

Note: An End of Block (EB) command must be used with every LU command.

The LU loop will continue to iterate until the specified conditional result is true. LU checks the conditional at the end of the loop block,therefore,the block is always executed at least once, even if the condition is true on the first iteration. Loop While (LW) defines loops where the conditional is checked at the beginning of the loop.

The standar d software allows up to 16 nested loops (one inside the other).A GT command within a LU loop will terminate the loop, clear the loop stack and jump to the new program. Following are examples of programs using LU:

Example #1: (A)=0 LU(A)=10 DI10 GO (A)=(A)+1 EB
In this example,the loop is executed 10 times with a final position 110 distance units.

Example #2: (A)=10 LU(A)<20 DI10 GO EB
In this example,the loop is executed once since the (A)<20 condition is true on the first iteration.

Example #3: LUXX1X1 MS1,"Inputs 3 & 5 are off" EB GT[Inputs On]
In this example,the loop will continue to execute as long as inputs #3 and #5 are off.

Example #4: LU4,1 MS1,"Input 4 is off" EB GT[Input On]
In this example,the loop will continue to execute as long as input #4 is off.

LW Loop While Condition True syntax - LW (See Below)

Units: n/a
Range: n/a
Default: n/a

The Loop While (LW) command defines a loop block in which loop iterations are based on a conditional result.The syntax for LW, which is identical to the IF command,is as follows:

Syntax: LUi,xx…
LUxx…
LU(Mathematical expression) or expressions (2)
where,
i = starting input number, 1-8 (SmartStep),1-16 (SmartDrive)
x = 0,(Input Off)
x = 1 (Input On)
x = anything else (Input level Don't Care)
Mathematical expression = Any valid conditional or logical expression

Note:An End of Block (EB) command must be used with every LW command.

LW will continue to iterate while the specified condition is true. LW checks the condition at the beginning of the loop block,therefore if the condition is false on the first iteration,the block is immediately skipped.Loop Until (LU) defines loops where the condition is checked at the end of the loop block.

The standar d software allows up to 16 nested loops (one inside the other).A GT command within an LW loop will terminate the loop, clear the loop stack,and jump to the new program. Following are examples of programs using LW:

Example #1: (A)=0 LW(A)<=10 DI10 GO (A)=(A)+1 EB
In this example,the loop is executed 11 times with a final position 110 distance units.

Example #2: (A)=10 LW(A)>20 DI10 GO EB
In this example,the loop is immediately skipped since the (A)>20 condition is false.

Example #3: LWXX1X1 MS1,"Inputs 3 & 5 are on"EB GT[Inputs Off]
In this example,the loop will continue to execute as long as inputs #3 and #5 are on.

Example #4: LW4,1 MS1,"Input 4 is on"EB GT[Input Off]
In this example,the loop will continue to execute as long as input #4 is on

MC Move Continuous — syntax - MC+,+

Units: n/a
Range: n/a
Default: n/a

MC sets move profiles to "continuous move", utilizing AC and VE parameters. Move Continuous is enabled on an axis with the "+" sign. MC+ enables the mode for axis one; MC,+ enables the command on axis two; and MC+,+ enables both axes. DI, DA and DC commands reset the mode to distance.

Each Move Continuous segment must contain a GO command. Accelerations and velocities may be changed in any segment. If no change is specified to one of these parameters, the last value will be used. It is not valid to issue positional commands (DI, DA, DC, GH, SP) to an axis while it is in a Move Continuous mode. However, you may make distance-based moves on the other axis while running one axis continuously. Any command is valid within an MC segment except Distance Commands (DA, DC, & DI).

The direction of the move is specified by the sign of the VE parameter. If the sign of the VE parameter changes between two segments, the control will automatically stop the motor (at the programmed rate) and change directions to the new speed. This makes changing directions based on variable inputs very easy to program using a scaled variable as the VE parameter.

Once a Move Continuous segment is started, it will continue to move at the speed specified by VE until either another VE is commanded, the ESC Key is pressed, or an End Of Travel, Kill Motion, Interrupt, or Stop Input is activated. A commanded velocity of zero (VE0) stops an MC move. Motion will also stop if you enter the Edit, Help, Copy, or Delete menus.

After a continuous move segment has started, command processing will continue when constant velocity is reached. Other commands are then processed sequentially.

This allows you to:

- Have asynchronous inputs change the speed of an axis
- Make motion profile changes based on time delays or input states
- Manipulate I/O while moving as a function of time, distance, or input states
- Change speed based on analog inputs or variables
- Have an operator update the speed of an axis through the keypad
- Servo to an analog input
- Make a one or two-axis joystick using analog inputs
- Start a continuous move on one axis, and make distance based moves on another

If a motor is making a move when it comes to the end of a program, the motor will continue moving, even after the program ends. This allows you to:

- Put different MC moves in different programs and select different speeds by running different programs.
- Change speeds based on Binary or BCD program select lines
- Call MC moves as subroutines
- Run from "hosted" RS-232C mode, where the computer commands speed changes
- Run another program from the keypad that does not violate MC syntax. So you could run another program from the keypad to change speeds, move the other axis, manipulate I/O, interface with an operator or calculate arithmetic.

Example 1: Basic Move Continuous syntax. Demonstrates how to change speed and stop MC moves based on time delays and input conditions.

```
MC+       Enable Move Continuous on axis 1
AC.1      Set the acceleration rate
VE50      Set top speed to 50
GO        Start the Move Continuous move, command processing will continue
          when axis 1 reaches constant velocity
TD2       Delay for 2 seconds at speed
VE25 GO   Decel to 25|
WT111     Wait for inputs 1,2, and 3 to go active
VE0 GO    Stop the move
```

Example 2: Demonstrates how to prompt an operator for speed changes on a single-axis SmartDrive. The move is started after the initial velocity prompt. The velocity only changes when the operator enters a new value via the keypad. The move can be stopped by entering a velocity of zero, or when any of the stop conditions defined above exist.

```
[One Axis MC]
MS1,"Enter the Velocity"     Prompt the operator
IV23,(V)                     Put the operator input in variable (V)
MC+ AC1
VE(V)                        Use operator-entered variable (V) as new speed
GO                           Change velocity of axis 1 to the new speed
GT[One Axis MC]              Repeat
```

Example 3: Demonstrates how to prompt an operator for speed changes on a two-axis SmartDrive.

```
[Two Axis MC]
MS1,"Enter the X Velocity"   Prompt the operator
IV23,(V1)                    Put the operator input in variable V1
MC+ AC1                      Enable MC and set acceleration on axis 1
VE(V1)                       Use operator inputted variable (V1) as new speed
GO                           Change velocity of axis 1 to the new speed
MS1,"Enter the Y Velocity"   Prompt the operator
IV23,(V2)                    Put the operator input in variable V2
MC,+ AC,1                    Enable MC and set acceleration on axis 2
VE,(V2)                      Use operator inputted variable (V2) as new speed
GO                           Change velocity of axis 2 to the new speed
GT[Two Axis MC]              Repeat
```

Example 4: Demonstrates the use of WT, OT and TD commands in continuous move.

```
MC+ AC3 VE3 GO               Start first segment
WT8,1 AC.1 VE10 GO           Wait for input 8 and change speed
TD5 AC.3 VE.2 GO             Wait for 5 seconds and change speed
WT3,1 VE-10 GO               Wait for input 3 and change speed and direction
OT11                         Turn on outputs 1 and 2
TD10 VE0 GO                  Wait 10 seconds and stop the move
```

Example 5: Demonstrates two-axis joystick control with a Smart Drive. The control must be equipped with two analog inputs (in this case we assume AI9 and AI10) and be connected to an external, user supplied joystick.

```
[Joystick]                      Main Program, X=Axis 1,Y=Axis 2
GS[Get Speeds]                  Call a routine that scales the analog input voltage
                                to X and Y speeds
MC+,+ AC.2,.2                   Enable continuous mode, set accelerations
VE(VX),(VY)                     Set speeds for X and Y axis
GO                              Start motion
GT[Joystick]                    When the axes reach constant speed, run
                                [Joy stick] and update the velocity based on the
                                current analog input voltage.

[Get Speeds]                    Converts two analog inputs to X and Y speeds
                                Returns variables (VX) and (VY), the X and Y
                                velocities
(Deadband)=.15                  Set desired zero speed deadband of joystick
(Neg Deadband)=-.15
(Max RPS)=5                     Set desired full deflection speed
(Scale)=(Max RPS)/28137         Scale factor based on Max RPS
(Offset X)=47685                X axis offset for analog in
(Offset Y)=50990                Y axis offset for analog in

(VX)=(AI9)-(Offset X)           Convert the analog input to a speed for X and Y
(VX)=(VX)*(Scale)
(VY)=(AI10)-(Offset Y)
(VY)=(VY)*(Scale)

IF(VX)<(Deadband)               If the speed is within the deadband, set the
                                speed to zero
IF(VX)>(Neg Deadband)
(VX)=0
EB
EB
IF(VY)<(Deadband)               If the speed is within the deadband, set the
                                speed to zero
IF(VY)>(Neg Deadband)
(VY)=0
EB
EB
EP                              Return to {Joy Stick} and update the speeds of
                                X and Y.
```

MS	Message to Display	syntax - see below

Units: n/a
Range: n = 1-40 display position in characters;Characters 1-20 on top line, 21-40 on the bottom display line.
Default: n/a

Syntax: MS,"" returns to the default runtime display
MSn,""
MSn,"user text"
MSn,(variable)

MS allows you to put messages on the keypad's display. Messages are typically used to prompt for operator input, display function key prompts, or as a diagnostic tool.

MSn,"" clears the display from the n^{th} character on. MSn,"user text" prints user text beginning at the n^{th} character. MSn,(variable) writes the value of the variable on the display beginning at the n^{th} character. The above variations to MS will disable the default position and I/O display until program execution stops. MS,"" can be used to restore the default axis position and I/O display during program execution.

Examples:

MS1,""	Clears the Display.
MS3,"Part Count"	Writes string beginning at character 3, top line.
MS27,(COUNT)	Displays the value of the variable (COUNT), beginning at position number 27 (7th character, 2nd line).

Below is an example of one segment of the Keypad Programming Template which can be found immediately inside the back cover of this manual. The template is provided to allow you to write MS programs by hand exactly as they will appear on the 40-character keypad display. Please feel free to make copies of the template for writing your programs.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M	S	2	7	,	(	C	O	U	N	T	)								
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
		FK1							FK2							FK3			

ON — ON Command (On Event) — syntax - ONn, GTx, ONn, GSx, ONn,0

Units: n/a
Range: n/a
Default: n/a

The ON command allows the user to define conditional program execution based on the occurrence of a certain event.When the programmable event occurs,the current program and move are interrupted and program execution begins at the predefined interrupt program.The interrupt program can be defined as a GT or a GS.Defining the interrupt as a GS allows program execution to return to the exact point in the original program where the interrupt occurred.The ON command currently supports EOT as an event conditional.

The syntax for defining an event and interrupt program number and type is ONn,GTx or ONn,GSx where n is the event type and x is the interrupt program number or name:

n:	L	On End of Travel Limit

The syntax for clearing a previously defined event conditional is ONn,0 where n is the event type as listed above and 0 (zero) is the clear event designator.

Once an ON event has been defined,it is active in all user programs and need not be redefined unless the user wishes to change the interrupt program type, number or name,or clear the event condition.

Example: Using ON to handle an EOT event

```
[POWERUP]
ONL,GT[ON EOT]          {Goto [ON EOT]on an end of travel input}
GT[HOME]                {Home the machine}

[MAIN]
LP                      {Loop infinitely}
 VE5 DA20 GO            {Define move}
 DA0 GO                 {Define move}
EB

[ON EOT]
IF(SA1)&256             {Check if EOT- switch was hit (bit #9 in SA)}
 VE1 DA0.5 GO           {Move off EOT- switch}
 GT[HOME]               {Home the machine}
EB
VE1 DA-0.5 GO           {Move off EOT+ switch}
GT[HOME]                {Home the machine}

[HOME]
GH3                     {Home}
GT[MAIN]                {Jump to main loop}
```

OT Output — syntax - see below

Units: n/a
Range: i = starting output number, 1-16
x = 0;output high (OFF)
x = 1;output low (ON)
x = anything but a 1 or 0 ; the state of the output remains unchanged
Default: n/a

Syntax: OTi,xx...
OTxx...(assumes first output is # 1)

Sets both discrete and digital Opto output states.Once an output is turned on,it will remain set until changed by another output command,a reset input (software warm-boot),or power is cycled.All outputs are turned off upon power up or during a reset.

To increase flexibility, the OT command allows you to use configured outputs anytime.To help prevent this added flexibility from causing programming confusion, you can use any character in the "don't change"section of your output statement. This allows you to self-document your OT statements. For example,assume you configured output #3 as a "FAULT"output.Programming like "OT01F10"can help remind you that you are already using output # 3.

Example:

OT4,1	Turn on Output 4
OT2,0D1	Turn Outputs 2 off, leave 3 as is,and turn 4 on
OT110	Turn Outputs 1 and 2 on,and 3 off

"" Quote — syntax - "Any ASCII character"

Units: n/a
Range: n/a
Default: n/a

The " " command transmits a string out the serial communications interface. A""without any string will transmit a carriage return character (ASCII 13).

Example:

"Move Complete"	Transmits string only out serial interface.
""	Transmits a carriage return only.

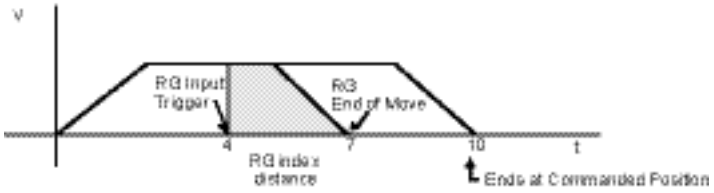
RG Registration — syntax - RGr,r

Units: n/a
Range: n/a
Default: n/a

The Registration Command (RG) specifies a distance to be indexed from the current position - as commanded by a specific input trigger. For example,in the following program of 10 user-units on axis #1,the input trigger is received at user-unit 4,to move 3 user-units from the point where the input trigger was received.

```
VE2 AC.1 DA10 RG3 GO
```

In the program above,assume the input was an optical sensor which triggered on a registration mark at a position of 4 user-units.The figure below shows the commanded move related to the registration move.



Accompanying the programmable Registration Command is the configurable Registration Input:G (also G in Serial Setup Commands).To configure a Registration input from the keypad, choose EDIT > SETUP > I/O > INPUTS.An input configured as a Registration Input will be designated by a G on the keypad input status display.The RG Command will only function if the corresponding input has been configured as a Registration Input (see note).

Application Notes

- Registration Input is only configurable on input #1 for axis #1,and on input #2 for axis #2.
- The RG command cannot be used for both axes during simultaneous 2-axis moves.When both axes are moving,only one Registation input will register and function correctly.

  Example:AC1 VE1 DI100 RG1 GI AC,1 VE,1 DI,100 RG,1 GI
  This program is syntactically correct,but only the RG,1 will work.

System Performance when Using the RG (Registration) Command

The input capture delay is 5µs. Worst case position error is 9 steps at 50 rps.

There is a Capture Window (Position Lag) associated with the RG Command,which is a function of move velocity and the Position Capture Delay (reaction time) and can be calculated with the following equation:

Capture Window = RG Position Capture Delay * Velocity (Steps/Sec)

The Capture Window value is the number of steps accumulated between the falling edge of the Registration input and the time the current position is captured. Depending on the FPGA hardware version of your Indexer this RG Position Capture Delay will either be 164 µs or 5 µs.

The Registration Command is only available on firmware version 4.0 & higher. If you have FPGA version 5.9 or earlier, the Capture Delay will be 164 µs.If you have FPGA 6.7 or higher, the position is captured in hardware,and the only Capture Delay is the input's opto-isolator (5 µs).Use the keypad's HELP key to determine your FPGA version.

Regardless of the FPGA version used, we have found both Capture Delays to be extremely repeatable.This leads to a very repeatable Capture Window (distance lag from when the registration input is made) that can be accounted for by decreasing your commanded registration distance by the Capture Window. For example, a motor traveling 240,000 steps/sec (30 rps with 8000 step/rev drive resolution) has a capture window of 39 steps for a 164 µs Capture Delay and 1 step for a 5 µs Capture Delay. Assuming the desired registration distance was 3 user units (assuming 1 user unit is 8000 motor steps),an RG2.9795 would result in the motor traveling exactly 3 user units for a unit with a 164 µs Capture Delay.

SP Set Position — syntax - SPn,n

Units: selected in EDIT > SETUP > MECH menu
Range: varies based upon Distance Units
Default: 0

SP sets the current absolute position to "n". This command is typically used to readjust or shift a coordinate system. It is often done after a series of incremental moves to reset the absolute coordinate frame.

Example:

MC+ GO WT1,1 VE0 GO SP10.5	After the move is complete, sets the current position of axis 1 to 10.5.

SQ Square Root — syntax - SQr,(var)

Units: n/a
Range: 0.00001 to 214748.3645
Default: n/a

The **SQ** command calculates the square root of a number and returns the result in a user defined variable. The n parameter in the syntax can be a number or a variable parameter, however, the second parameter must be a previously defined variable for which the square root result is stored. If the second parameter is not a defined variable, you will get a **Bad Variable Name** error. Following mathematical convention, SQ will produce an **Invalid Parameter** error for negative "r" values. The return value is accurate to the 0.01 place.

Example: The following example program calculates the square root of 27.96 and stores the value in the user defined variable (SQRESULT).

Program: (SQRESULT)=0 SQ27,(SQRESULT)
The returned value in (SQRESULT) would be 5.28.

ST — Stop On Input — syntax - STn,n or ST#n,#n

Units: n/a
Range: 0-16 (Inputs)
#1, #2 (Axes)
Default: n/a

Syntaxes:
ST stops move execution upon activating the input specified by n.
ST0 disables (turns off) the STn command.
ST#1 stops move execution on axis #1.
ST,#2 stops move execution on axis #2.
ST#1,#2 stops move execution on both axes.

ST#n functions identically to the STn command without the use of an input allowing program command conditional motion termination.

After the ST command is executed, the specified input is monitored during every "move profile." If the imput is activated, the current "move in progress" is terminated, stoping all motion until the input is deactivated or a ST0 is processed. the drive will process and calculate commands, but it will wait at the next GO command until the ST input changes.

The motor is stopped at the deceleration rate specified in the Stop Decel Rate setup parameter. Once issued, Stop on Input remains active until it is turned off by the ST0 command, a reset is issued, or power is cycled.

Example:Move to absolute position of 6 distance units. If (A)>10, motion is stopped.

```
AC,1 VE25,25 DA6,6 GI        {Move to 6 absolute distance units and Go
                              Immediate}
IF(A)>10                     {Check value of A (assume A was previously
                              defined}
ST#1 TD1 ST,#2               {Stop motion on axis #1 wait 1 sec. then stop
                              axis #2}
EB
ST#1,#2                      {Stop motion on both axes}
```

TD — Time Delay — syntax - TDr

Units: seconds
Range: r = .01 to 99999.99 seconds
Default: n/a

Delay r seconds before executing the next command.

Example: VE50 DI4 GO OT11 TD.5 OT00

Move 4 units, turn outputs 1 and 2 on, delay .5 seconds, and turn outputs 1 and 2 off. See also: System variable (TIME), in Chapter 7, Programming Your Application.

VE Velocity — syntax - VEr,r

Units: selected from the EDIT > SETUP > MECH menu
Range: .002 - 50 rps. Range is scaled to velocity units
Default: 1 motor rev per sec (rps)

VE sets the maximum velocity during a move profile. If the acceleration rate is not high enough or the move distance is not long enough the motor may end up making a triangular (velocity vs. time) move and the motor may never reach the specified speed. Once VE is specified, the value is used in all subsequent moves until re-defined.

Example:

AC.1 DE.2 VE50 DA4 GO	Move to absolute position 4 units with a top speed of 50 units/sec.

WT Wait — syntax - see below

Units: n/a
Range: i = starting input number, 1-16
x = 0; input high
x = 1; input low (grounded)
x = anything but 1 or 0 ; ignore the input level

expression = any valid expression as defined in the math and variables section.

Default: n/a

Syntax: WTi,xx...
WTxx... (assumes first input is input 1)
WT,expression
WT#1, used with GI moves

This command waits for the specified condition to be true before continuing execution of a program. Either digital or analog input conditions may be used.

To increase flexibility the WT command allows you to use configured inputs in the expression. To help prevent this added flexibility from causing programming confusion, you can specify any character as an input (x). This allows you to self document your WT statements. For example, assume you configured input #3 as a "JOG SPEED" input. Programming like "WT01J10" can help remind you that you are already using input # 3.

Example:

WT4,1 GO	Wait for input 4 to equal 1 before moving
WT2,010 GO	Wait for inputs 2-4 to equal 010 before moving
WT110 GO	Wait for inputs 1-3 to equal 110 before moving
WT#1	Causes program execution to halt until GI move is complete

Note: In order to synchronize program execution with the end of a GI move, there is new syntax associated with the WT command: WT#1,#2 will halt program execution until the respective axis has completed its move. WT#1 will wait for only axis #1; WT,#2 will wait only for axis #2; WT#1,#2 will wait until both axes have stopped.

Summary of Operators, Functions and Expressions	
[ ]	Name Program
( )	Name Variable
&&	Logical AND
||	Logical OR
!	Logical NOT
!=	Not Equal
+	Add
-	Subtract
*	Multiply
/	Divide
=	Equal
>	Greater Than
<	Less Than
>=	Greater Than or Equal to
<=	Less Than or Equal to
&	Bitwise Boolean AND
|	Bitwise Boolean OR
++	Increment Variable
+=	Increment by n
--	Decrement Variable
-=	Decrement by n
<<	Shift Left
>>	Shift Right

Refer to Chapter 5, Programming Your Application, for more details and examples of how to use the above operators, functions, and expressions.

Programming with Serial Communication

Overview

Any RS-232C terminal,PC,computer serial RS-232C card,or RS-232C-equipped PLC can be used to configure,program,and operate IDC's SmartDrive controls.IDC provides and strongly recommends using our Windows-based Application Developer for configuration and programming.If you choose not to use this tool,all of the IDeal™ RS-232C programming and setup commands are listed alphabetically later in this chapter.

Application Developer provides a graphical control configuration environment, a program development editor, and a terminal communication package. Application Developer also provides application upload and download utilities,and an on-line software reference help utility.

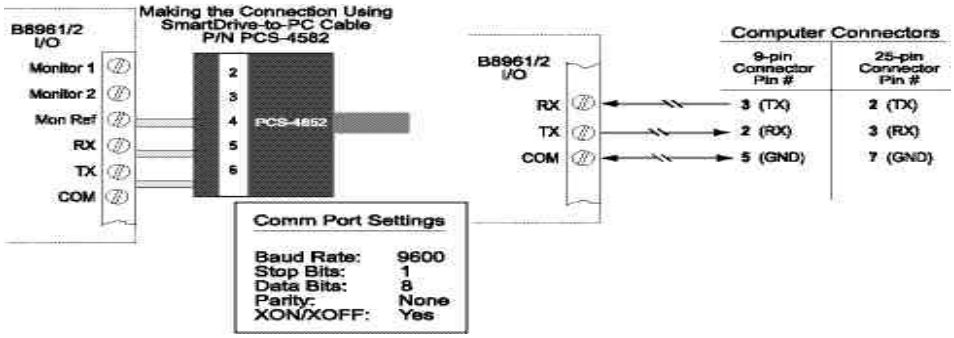
This chapter is divided into 4 main sections.Section 1 covers common RS-232C details including baud rate settings as well as hardware and daisy chaining information.All RS-232 users will need this information.Section 2 covers the installation of Application Developer. Section 3 covers Using Application Developer to setup and program SmartDrive systems.Section 4 provides details on all of the RS-232C setup commands that Application Developer employs.This section will be useful to users who are not using Windows,or who plan to run the SmartDrive in a "hosted"environment.(i.e.the host streams down individual commands for immediate execution,or calls previously defined programs.) The host could be a PC,RS-232C equipped PLC,or some other type of intelligent device.IDC recommends that even users whose final application will be run in hosted mode use Application Developer to initially configure their system.

Section 1: RS-232 Protocol

IDC's SmartDrive series controls use a 3-wire implementation of RS-232C.The RX,TX, and COM lines are the serial signals supported. No hardware handshaking is required. Note that some RS-232C devices require handshaking, such as RTS and CTS. It is the responsibility of the user to disable this handshaking via software or hardware.

Making RS-232C Connections to the B8961/2

Make your RS-232C connection as shown below.The SmartDrive-to-PC Cable (IDC P/N PCS-4852) is an ideal choice for making trouble-free connections.



Troubleshooting Serial Communication Problems

If communications between computer and SmartDrive are unsuccessful, one or more of the following procedures will usually solve the problem:

1. Test your terminal or terminal emulation software.Unhook the drive and transmit a character. An echoed character should not be received.If an echoed character is received, you are in half duplex mode. Jumper your host's TX and RX connections,and transmit another character. If you do not receive the echoed character, consult the manufacturer of the host's serial interface for proper pinouts.
2. Host transmit (TX) must be connected to receive (RX) of the drive unit,and receive (RX) of the host must be connected to transmit (TX) of the drive.If communication fails,try switching connections on either the host or the drive.
3. Many serial ports require handshaking. Jumper RTS to CTS,and DSR to DTR (see table).

Jumpers	9 pin D	25 pin D
RTS to CTS	7 to 8	4 to 5
DSR to DTR	4 to 6	6 to 20

4. Configure the host to the identical baud rate, number of data bits, number of stop bits,and parity.
5. Receiving double characters (XX) when entering single characters (X),indicates your computer is set to the half duplex mode.Change to the full duplex mode.
6. Check your grounds.Use DC common or signal ground as your reference.Do not use earth ground or shield.
7. Check your cable length.If any cable is over 50 ft.long, you should be using a line driver, optical coupler, or shield.Shields must be connected to earth ground at one end only.

Daisy Chaining SmartDrives

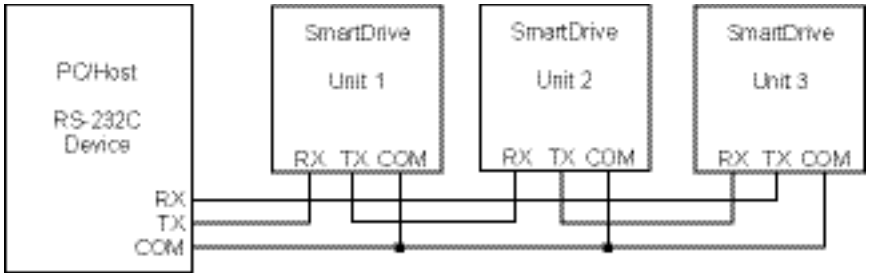
Your SmartDrive also supports daisy chaining.The unit address (range 1-99) can be set via the keypad,through Application Developer, or with a terminal program using the Unit Number (UN) command,or the entire chain may be addressed at once using the Auto-Address (AA) command.

Rules for Daisy Chain Operation

1. Units on a daisy chain must be "device addressed'(numbered) in ascending order away from the host device/controller in order for the Load All (LA - EX) commands to work properly.The unit addresses are not required to be numerically sequential,but must be in ascending order.
   Example:1,2,4,6,8 is valid addressing. 6,3,10,8,2 is not valid.
2. Do not duplicate unit numbers or addresses.
3. RS-232C Echo should be enabled for each unit on the daisy chain.Disabling RS-232C Echo will prevent the daisy chain from functioning properly.
4. Any loose RS-232C connections or miswiring along the daisy chain will cause communication to fail.Please double check wiring if communication problems arise.
5. "Device Addressing"RS-232C commands (using the specific unit number in front of the command) is necessary if the user wants only one specific unit to perform an operation.
6. Status commands require addressing.

Please call IDC if you need to daisy chain more than 99 drives.

The hardware configuration below shows how to connect the daisy chain.



Section 2: Application Developer Software

All of IDC's SmartDrives come with Application Developer software.The programs and data files are automatically installed with a setup utility included on the disk.

Installing Application Developer in Windows 95, 98, or NT

1. Insert Application Developer CD-Rom in your CD-Rom drive.
2. Click on Start.
3. Click on Run.
4. If there is no drive specified in the Command Line box,type the drive letter, followed by a colon (:),then a backslash (\),and the word **setup**.
   For example: **d:\software\Application Developer\setup**
5. Click OK.

The installation program will display the Welcome screen shown below. Throughout the installation process,the program will prompt you to enter information.Click on the **Next** button to continue the installation.At any point during the installation, you can click on **Cancel** to abort the process.

The Setup program will take you through a series of screens to verify user information and destination directories.

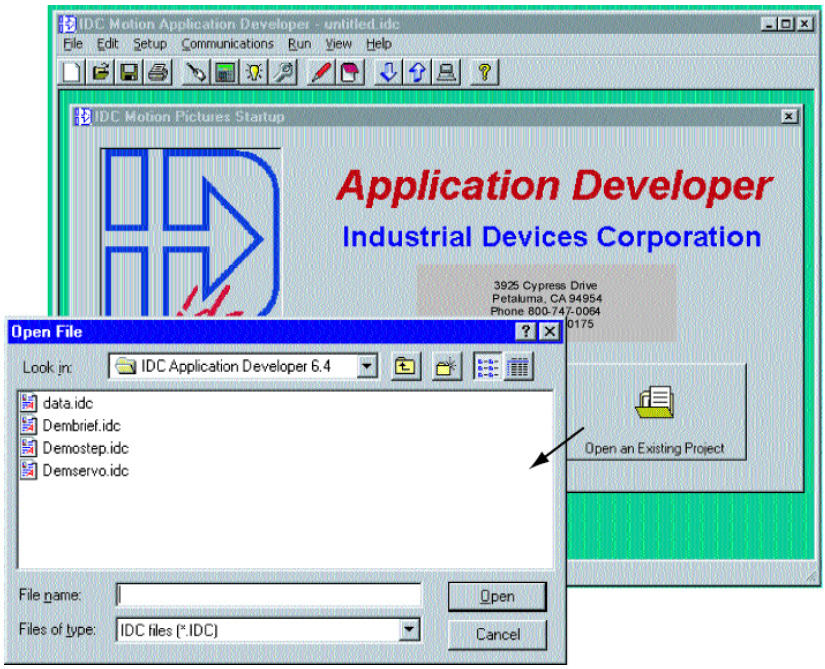


Common Installation Errors and Remedies

There are no known installation problems with Application Developer. Please call the IDC Applications Department at (US 1-800-227-1066) (INT'L 1-603-893-0588) (from outside U.S.) if you need assistance installing this software.

Section 3: Using Application Developer

Application De veloper's graphical environment helps you setup and program your SmartDrive from your computer. It guides you through configuring your control, following the same steps and menus as the keypad configuration.Complete details on setup menus and choices can be found in Chapter 5 - Configuring Your System . Applications (programs and configuration files) may be created,saved,edited and downloaded (sent) to your control. Application De veloper will also upload (receive) an entire setup and program memory from a control.The window shown below appears immediately after starting the Application Developer program.



Upload Project

To begin immediately with a drive,select Upload Project and then the controller type.Com Port setting will be confirmed before uploading from the drive or keypad (depending on controller type).

Launch Now

Launch now allows immediate software launch, bypassing Uploading,opening an existing project and project wizard.The controllable type will be prompted.

Project Wizard

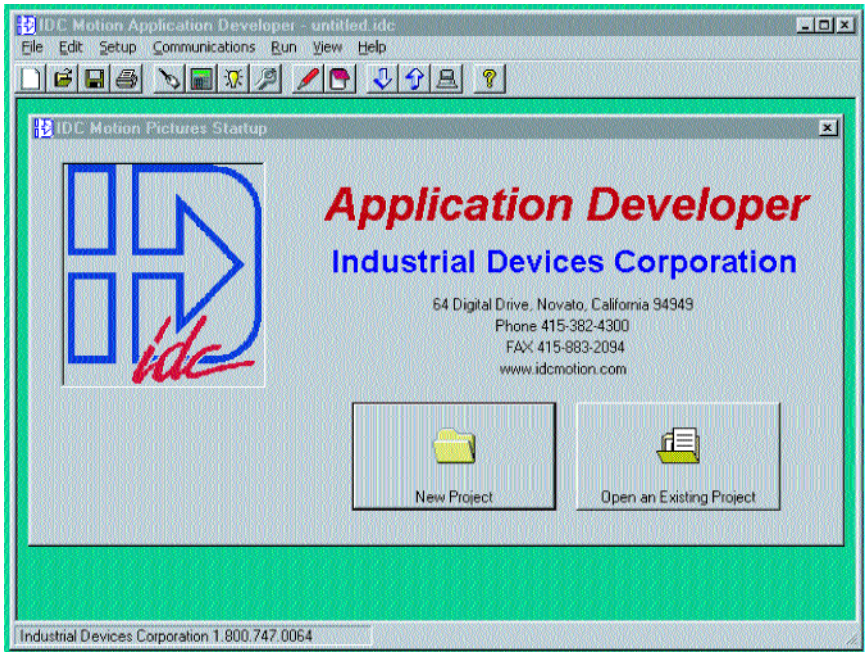
The Project Wizard allows the quickest and most accurate initial selection of drive, encoder, mechanics,and motion units.This chapter contains information to help you get started using Application Developer.

Open an Existing File

If you "Open an Existing Project",Application Developer will bypass the Setup Wizard and allow you to select a file in the Open File window (see below).After you have opened an existing file it is still possible to use the Setup Wizard at any time by simply clicking on the Wizard tool bar button.

New Project - Using the Project Wizard

1. Click on the **New Project** button to start the Setup Wizard.



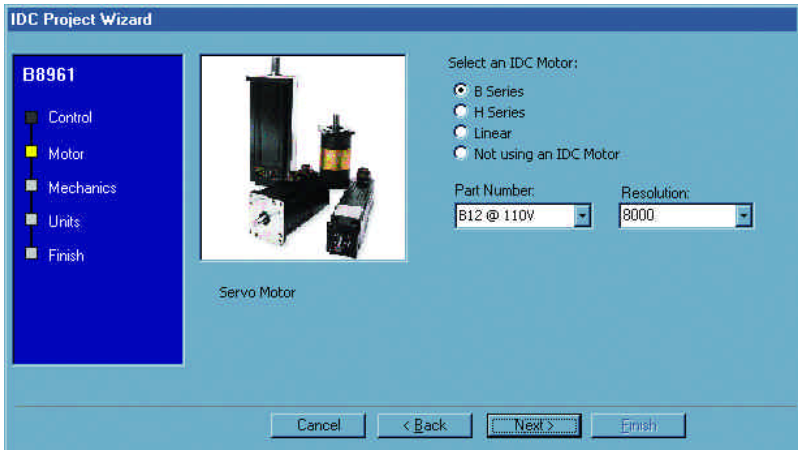
2. Select the product you are configuring, **S6962** in this case,and click on the **Next** button.

   Note:an a B8962 will be configured in this example of using the Project Wizard.If you are configuring an B8961, you will only be configuring one axis.

   The **Wizard Navigator** (left of photo) allows you to quickly locate your current position at any time during the setup process.The column of boxes represents the axes to be configured.A box's color represents its configuration status,i.e. **gray** indicates a parameter that has not been configured, **yellow** indicates the current position of the Wizard (parameter being configured),and **black** indicates that the parameter has been configured.



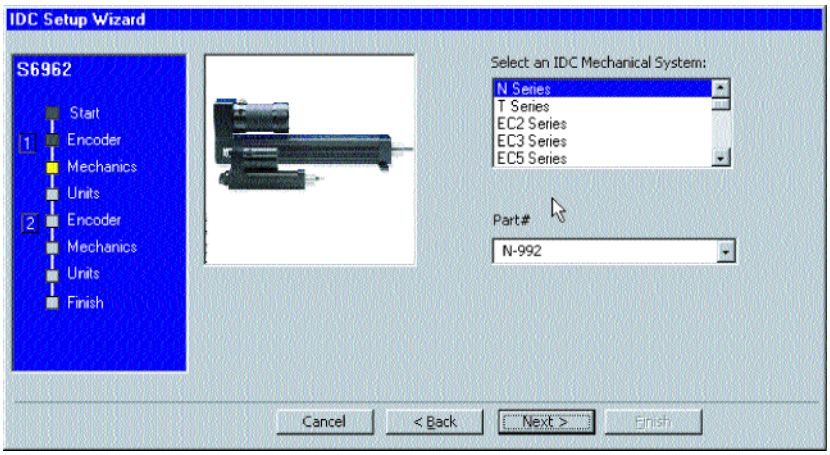
3. Click on the Next button and the Motor setup window will appear.



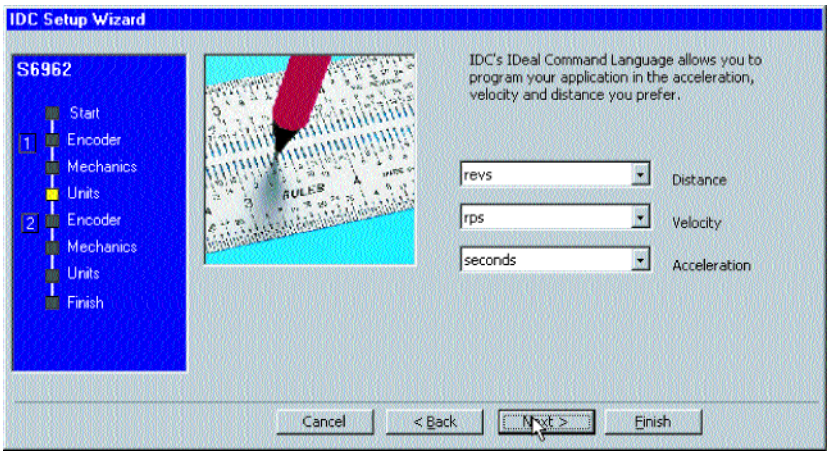
4. Select the motor and AC voltage that will be used,and select Next.

5. Click **Next** and you will be in the Mechanics setup window. Mechanical System setup is particulary easy if you are using IDC systems.

   Select the Mechanical System and Part # you intend to control with this axis, and the Wizard will calculate the rest.Select Non IDC Mechanics if you are using a non-IDC system,or motor only.

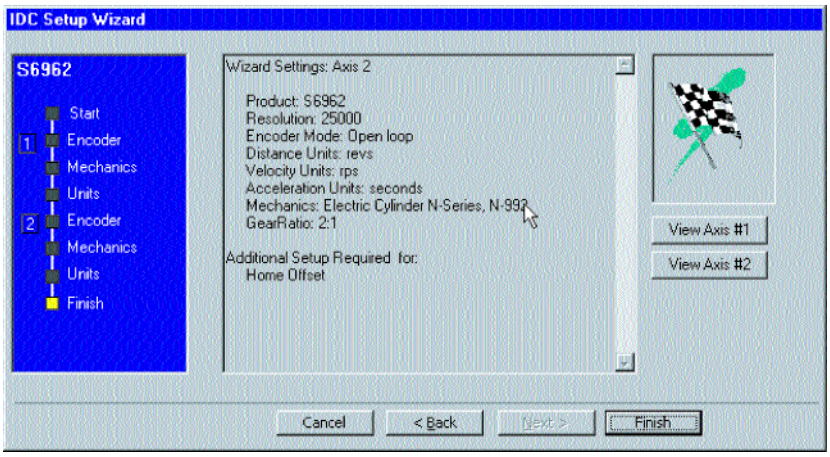


6. Clicking on the **Next** button will bring up the **Units** setup window.The SmartDrive lets you program the control in the units that work best for your application.



7. Select your preferences from the pulldown menus and click on **Next**.

8. When all axes have been configured,the following window will appear.This display gives you the opportunity to review the settings you have made for each axis, and to return (using the Back button) to any parameter you may wish to change at this time.



9. When you are satisfied with the setup of your system, click on **Finish**. Note that the Wizard Navigator now indicates that all axes and parameters have been configured.

After Using the Setup Wizard

Your basic system setup parameters have been configured by the Setup Wizard,and though the system is functional, your application will probably require further configuration and fine-tuning.

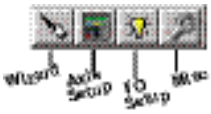
The remainder of this chapter provides the same setup information as configured by the Setup Wizard plus more detailed information on each parameter to allow more control of your system and greater capacity for applications.

The Setup Wizard remains available at any time by clicking on the **Wizard** button. The **Axis Setup**, **I/O Setup**, and **Misc** toolbar buttons provide access to the setup parameters that will be explained throughout the remainder of this chapter.

Setup Tool Bar Buttons

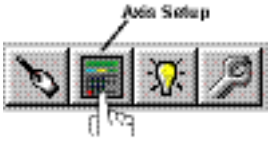
To access the setup windows and menus,simply click on the desired toolbar but-

tons as shown here:



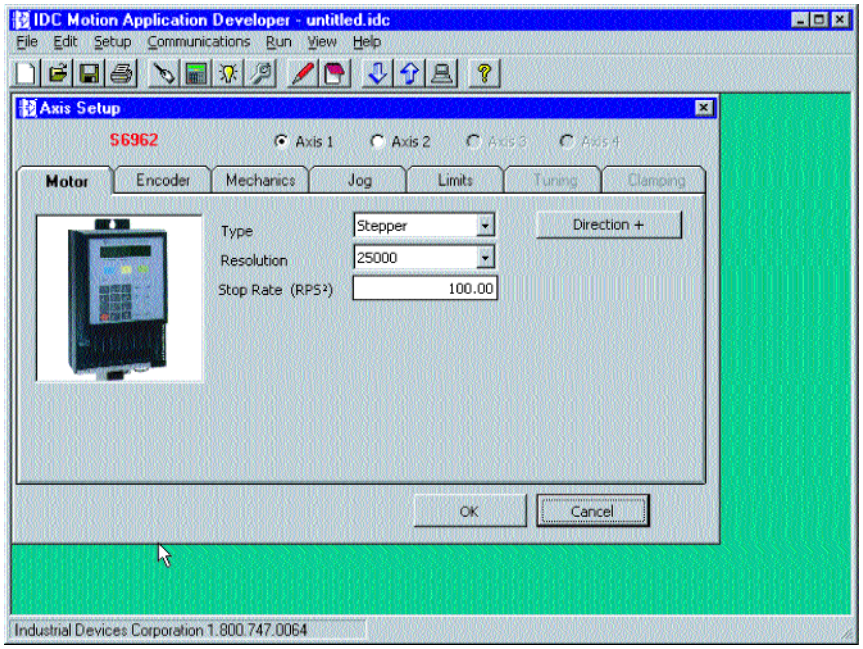
Axis Setup

Click on the Axis Setup button.



The first menu to appear will be the Motor menu,as shown below:

Note: please refer to Chapter 5 for detailed descriptions of setup parameters.



Axis

Each axis of your system must be configured separately.

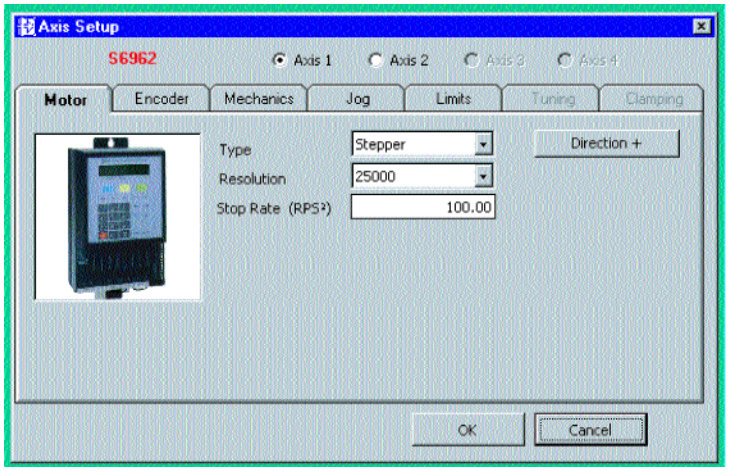


1. Select the axis to be configured.
2. Configure parameters for each axis as it applies to your application (Motor, Encoder, Mechanics, Jog,Limits).

Motor Menu

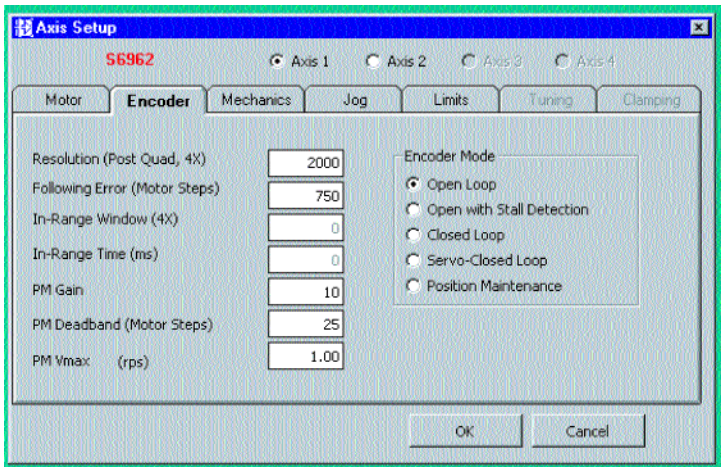
Settings for Drive Type,Resolution,Stop Decel Rate,and motor Directions are

selected in the **Motor** menu.



Encoder Menu

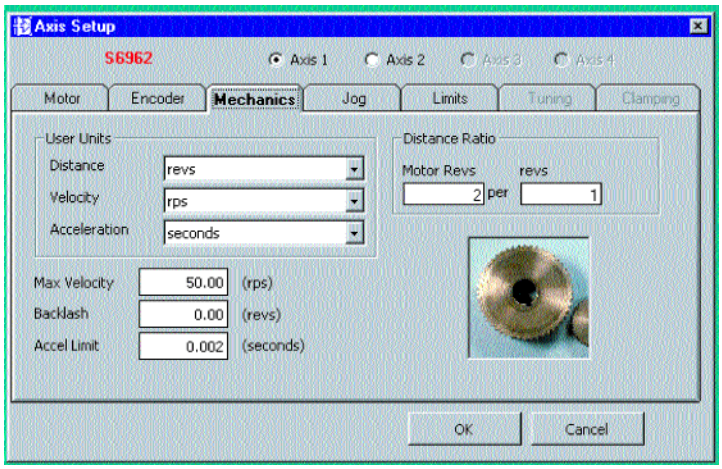
The encoder will already be defined based on the motor selected.The mode is fixed to Servo-Closed Loop.The Resolution and Mode cannot be changed.All others will have defaults,and shouldn't be changed unless using tuning software.



Mechanics Menu

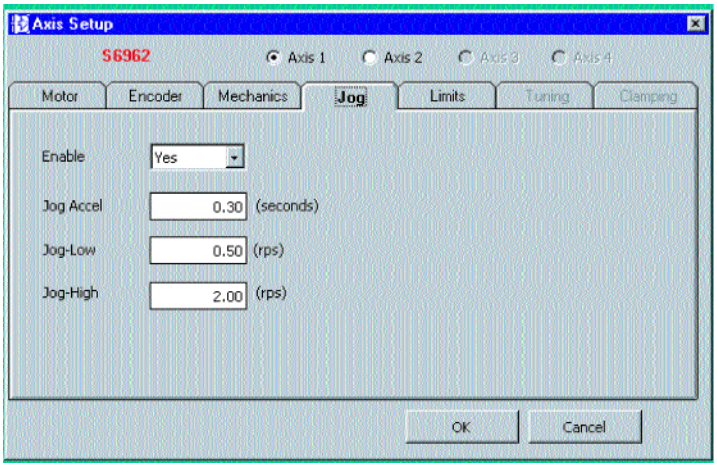
The Mechanics menu allows you to program distance, velocity, and acceleration

units convenient for your application.This menu also allows you to set a maximum allowable speed and acceleration for each axis.If programming in steps, gear ratio will be fixed 1:1.If programming in revolutions, revs/sec is in terms of the output revs (using the Distance Ratio as the gear ratio);rps is in terms of the motor revs.



Jog Menu

The parameters which control your jog operation are configured using the Jog menu shown below. If programming with acceleration in rate (rps2 or rev/sec2), change Jog Accel to appropriate value.

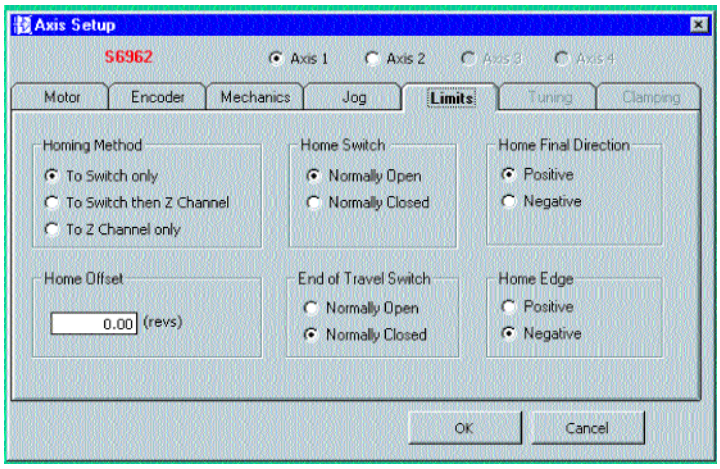


,

Limits Menu

Your B8961/2 has a built-in homing function which combines the flexibility of a

customized homing routine with the ease of use of calling a “canned”program.Also see the GH command in the IDeal Command Reference chapter for more details on homing.



Tuning Menu

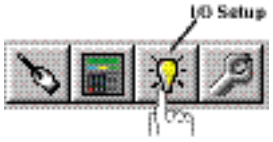
The tuning gains can quickly be changed in this menu.Default gains are input based on the motor selected.It is not recommended to change these values unless using the tuning software.

Clamping Menu

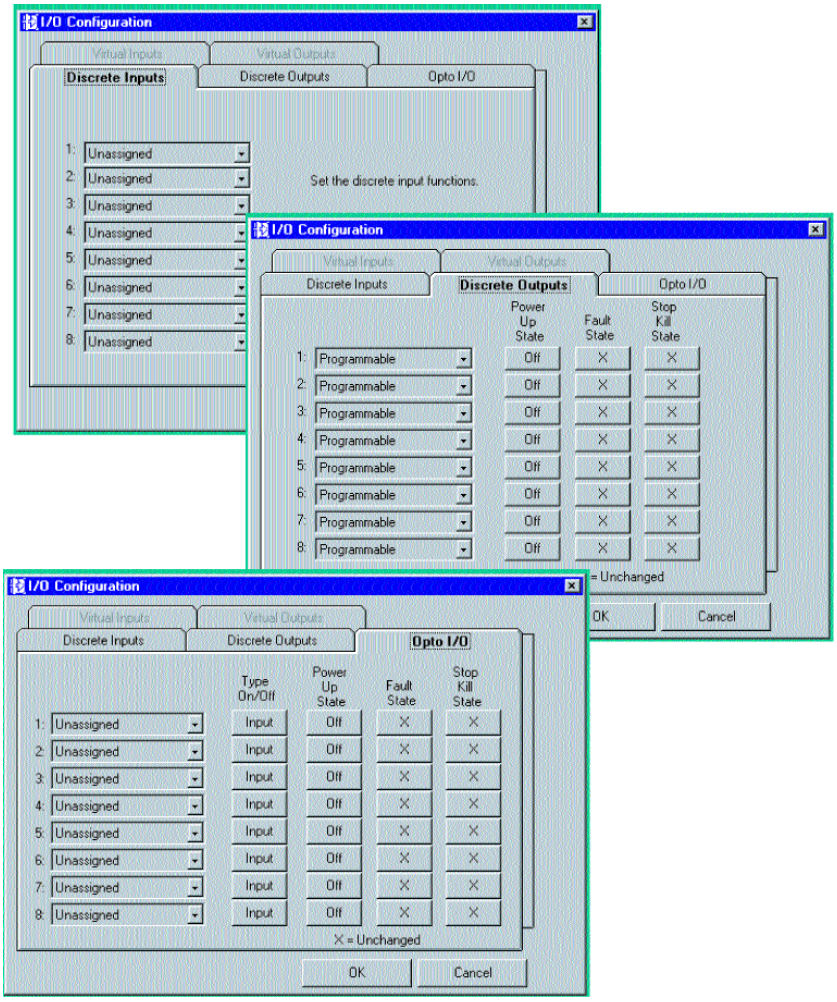
The Break Velocity and Clamping units may be selected in this menu. For further description,see the CL/CT section.

I/O Setup

Click on the I/O Setup button.

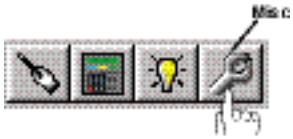


To define a dedicated function for each input and output,scroll through the pull-down lists and select from the available choices.



Miscellaneous (Misc) Setup

Click on the Misc (Miscellaneous) button.



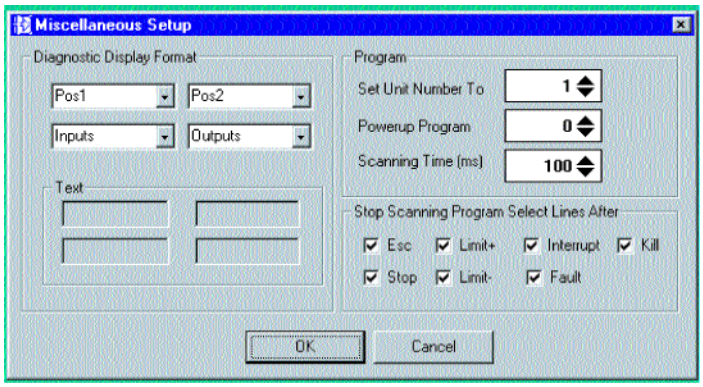
Misc Setup contains three configuration categories that include options available in the PROGRAM SETUP and RS-232C SETUP menus detailed in Chapter 5.

Diagnostic Display Format

Allows default display to be changed to position, velocity, blank,current,inputs,outputs,OPTOs,status (axis or system) or user defined text.

Program

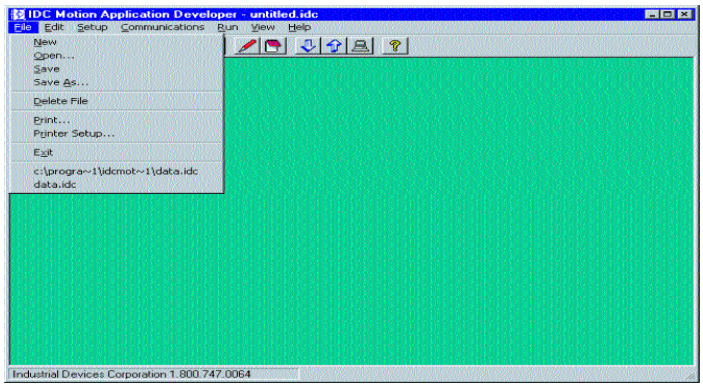
a. Sets the SmartDrives address.
b. Sets the program to run on power-up. No program will run if set to 0.
c. Sets the debounce (scan time) time of the program select inputs.



Stop Scanning Program Select Lines After...
Allows you to select the conditions under which program select input scanning stops.Scanning will resume upon cycling power, sending serial reset command or asserting a Warm Boot defined input.

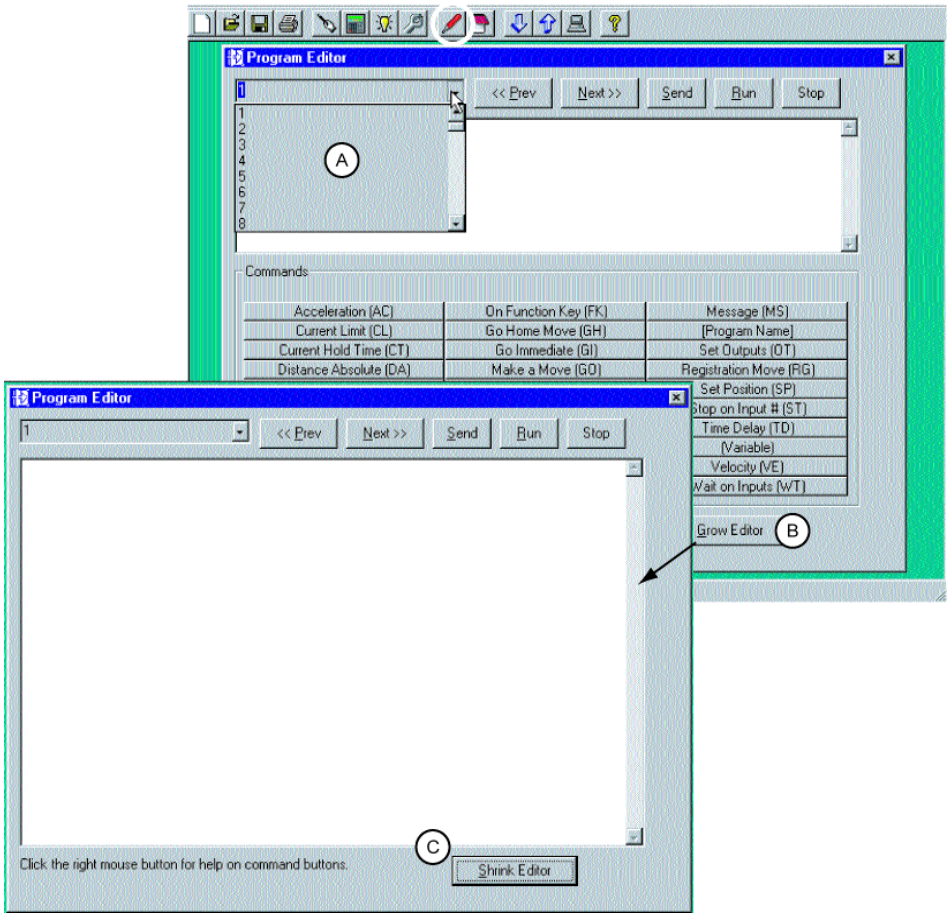
File Menu

Applications (programs and configuration files) may be stored on disk as DOS files. The default suffix is *.idc.The other selections under File are generic to all “Windows”applications.The *.idc program file is a configured text file.It can be viewed in a text editor program as well,or viewed in the View Configuration Text File.



Program Editor

The **Program Editor** features the standard windows editing features,Cut,Copy, and Paste,plus Send,Run and Stop.Undo,Delete and Select All are available by pressing the right mouse button,and an online HELP file may be reached via the F1 key or selected at the bottom of the editor.

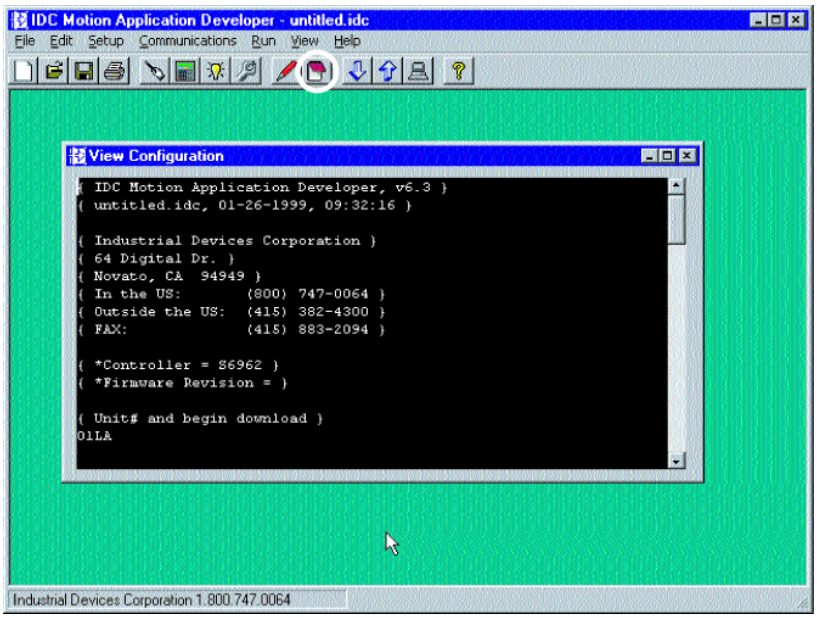


The drop down menu box (A) in the upper left hand corner shows the number and name of the currently active program,plus a list of up to 399 more programs. When the entire file is downloaded to the SmartDrive,these program numbers correspond to the program numbers the controller uses for binary and BCD program selections.

Program comments are placed between brackets {comments}.These comments are not downloaded to the SmartDrive Total program length, not including comments, is limited to 1024 bytes.Total file length without comments is 8k for B8961 or B8962, or 30k for B8961-30k or B8962-30k (expanded memory option).The Send button will send the viewed program (only one viewed) to the drive. Run will command the drive to execute the viewed program and Stop will halt program execution.

View Configuration

Click on the View Configuration button at any time to see your system configuration status.All configuration parameters are listed and may be viewed by scrolling the list.



I/O and Status

Drive and input/output/opto status can be displayed by selecting View → Status. A terminal window pane is available showing serial commands and drive response. Drive status response is hexadecimal code (see SAn,SS,SD serial commands for further details). Poll...will constantly update the status windows,where Refresh will update the status window once.

Communications

All Communications functions may be accessed from the main menu bar. Send All, Receive All and IDC Terminal are selectable on the Toolbar as shown here:



Send All

Use Send All to download the application you have developed.In addition to motion programs, your application file will include the setup commands derived from the choices you made in the Setup dialog boxes. Send All completely configures the SmartDrive control,and will overwrite any existing programs or configurations in the control.The feature allows easy configuration of repeat machines.Program comments will be stripped off before being sent to the SmartDrive.IDC recommends saving the commented version of your application before downloading.

Retrieve All

Use Retrieve All to upload the entire contents of a SmartDrive control to a new file that can then be edited,downloaded to another SmartDrive,or saved to a PC file for documentation purposes.This file contains the complete contents of the SmartDrive including all the programs defined,I/O definition,and mechanical scaling parameters.

Please note that this version of your application does not contain any comments, as they are stripped off during download to conserve memory in the SmartDrive.

Change Unit Number

Change Unit Number is used to set the device address of the control that Send/Receive Program uploads and downloads to on a single RS-232C daisy chain. Each unit must have its own unique software address.The Unit number of each control should be set BEFORE the units are connected in a daisy chain (the default address is one),or use Auto Address button.See RS-232C Protocol, earlier in this chapter, for hardware information on daisy-chain wiring.

Setup Password

An operator password or administrator password may be set or erased from this window.

Setup Comm Port

Comm Port is used to select a Comm Port when your PC has multiple serial ports. This dialog box also has a comm port test utility to verify proper RS-232C operation.

IDC Terminal

Terminal is a standard terminal emulator used for on-line communication with a SmartDrive control.It is very useful for troubleshooting interactive host/control communications.Caps lock needs to be on.Every character typed will be sent to drive;if mistyped,press enter and start again.All IDeal commands can be entered (Serial Programming,Serial Setup,Serial Immediate and Serial Supervisory).

Run Menu

Run > Program is used to run a specific program from Application Developer. Programs can also be initiated via dedicated program select inputs,through the keypad,or via any terminal using the RN command.

Run is only accessible from the main menu bar.

Section 4: RS-232C IDeal™ Command Reference

Overview

Though we strongly recommend taking advantage of the capabilities and convenience of Application Developer, you can configure, program, and run a SmartDrive from any RS-232C terminal or computer. If you do not intend to use the Windows software tools we provide, you will need to use the IDeal™ RS-232C command listings that follow.

"Hosted" or "interactive" motion control from a PLC or PC is also a common mode of operation. You can write your control programs in your language of choice (BASIC, C, ladder, etc.).

RS-232C SmartDrive operation is divided into four categories of commands. The first category is ***Serial Setup Commands***. These are the commands that IDC's Application Developer program uses to configure the SmartDrive according to the choices made in the SETUP dialog boxes. These ***Serial setup Commands*** include the syntax of the command, but the full command definitions and examples are found in Chapter 5 - Configuring Your System .

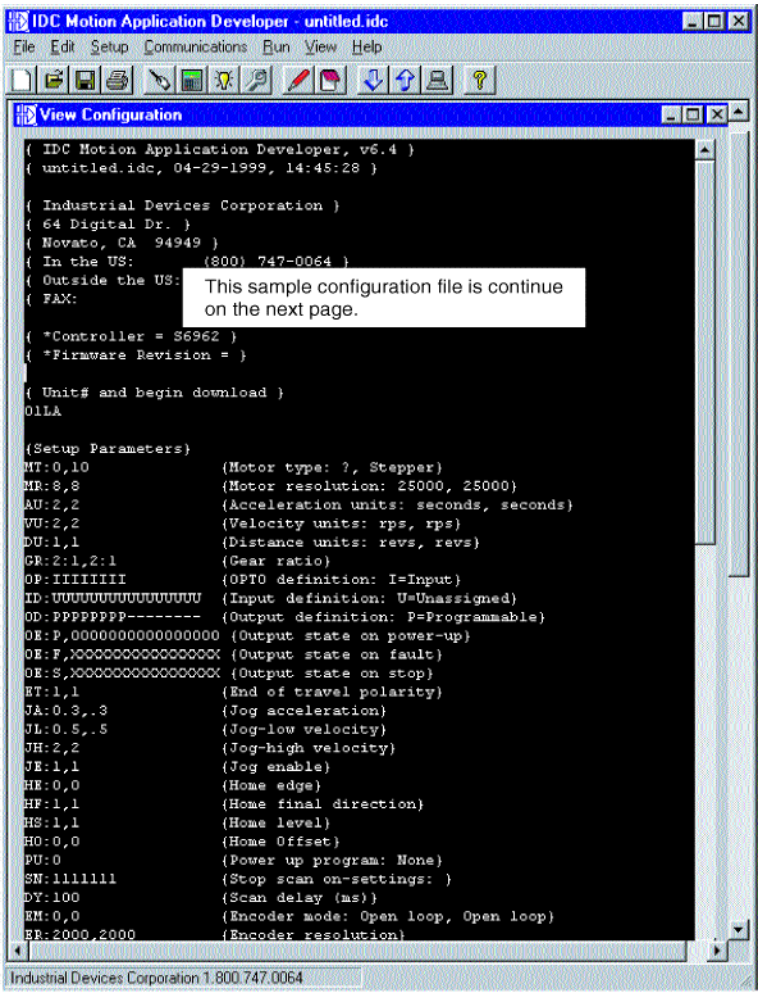
The second category is ***Serial Programming Commands***. These are commands that can be executed over RS-232C or downloaded to the SmartDrive's non-volatile memory for later execution. This category of commands is identical in syntax and functionality to the equivalent keypad command. These run-time RS-232C commands are listed in the RS-232C Command Reference, but the full definition and examples are listed in Chapter 6 - Programming Commands .

The third category of Serial commands is ***Serial Immediate Status Commands***. These commands bypass the normal command buffer and are executed immediately, regardless of what else the SmartDrive has been asked to do. These commands include instantaneous position reporting, drive status, and emergency kill and stop commands.

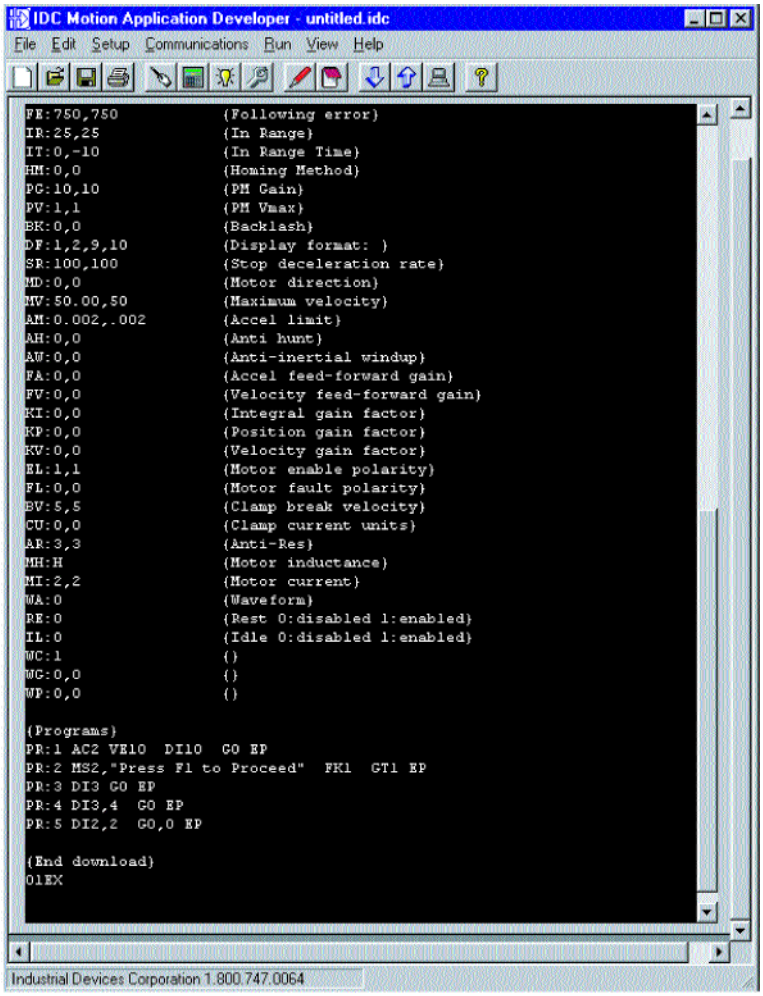
The final category is ***Serial Supervisory Commands***. These are the actual uploading and downloading of the setup and program parameters. Once an application with setup parameters and command has been created, these commands are used to download and upload the file to and from the SmartDrive.

Sample *.idc File

To familiarize yourself with IDeal RS-232C commands, review the following example of a typical file created in Application Developer for download to a B8962. You will need to generate a similar file to configure your SmartDrive. Individual programs can be downloaded in such a configuration file, or downloaded separately at run-time. The SmartDrive can also execute run-time commands in an "immediate" mode outside of any program. See the Command Syntax section of this chapter for more details.


IDC Motion Application Developer - untitled.idc
File Edit Setup Communications Run View Help
View Configuration
{ IDC Motion Application Developer, v6.4 }
{ untitled.idc, 04-29-1999, 14:45:28 }
{ Industrial Devices Corporation }
{ 64 Digital Dr. }
{ Novato, CA 94949 }
{ In the US: (800) 747-0064 }
{ Outside the US:
{ FAX:
This sample configuration file is continue on the next page.
{ *Controller = S6962 }
{ *Firmware Revision = }
{ Unit# and begin download }
01LA
{Setup Parameters}
MT:0,10 {Motor type: ?, Stepper}
MR:8,8 {Motor resolution: 25000, 25000}
AU:2,2 {Acceleration units: seconds, seconds}
VU:2,2 {Velocity units: rps, rps}
DU:1,1 {Distance units: revs, revs}
GR:2:1,2:1 {Gear ratio}
OP:IIIIIIII {OPTO definition: I=Input}
ID:UUUUUUUUUUUUUUUU {Input definition: U=Unassigned}
OD:PPPPPPPP-------- {Output definition: P=Programmable}
OE:P,0000000000000000 {Output state on power-up}
OE:F,XXXXXXXXXXXXXXXX {Output state on fault}
OE:S,XXXXXXXXXXXXXXXX {Output state on stop}
ET:1,1 {End of travel polarity}
JA:0.3,.3 {Jog acceleration}
JL:0.5,.5 {Jog-low velocity}
JH:2,2 {Jog-high velocity}
JE:1,1 {Jog enable}
HE:0,0 {Home edge}
HF:1,1 {Home final direction}
HS:1,1 {Home level}
HO:0,0 {Home Offset}
PU:0 {Power up program: None}
SN:1111111 {Stop scan on-settings: }
DY:100 {Scan delay (ms)}
EM:0,0 {Encoder mode: Open loop, Open loop}
ER:2000,2000 {Encoder resolution}
Industrial Devices Corporation 1.800.747.0064

Sample configuration file continued:



Command Syntax

All IDeal™ commands use two letter UPPER CASE ASCII characters. Command delimiters can be a carriage return (<cr>) or space (<sp>) character.

The commands that follow define IDC's command syntax. A brief command description is given here,but the full command definition is found in Chapter 5. This listing is intended only to help programmers with command syntax. IDeal programming commands are defined in Chapter 6.

The table below describes the abbreviations and format used in our command syntax definitions:

Letter or Symbol	Description
<n>	Unit address number is optional with RS-232C, and the command is sent to all units if no address is specified. All Status commands require an address. Example: <n>**AUi,i**
,	Field separator.
a	Alphabetic character.
h	Hexadecimal number.
i	Decimal integer number.
r	Decimal real number (up to 4 places to the right of the decimal.
:	A colon (:) is a neutral character. It can be used in a command to make it more readable to the programmer. For example OP:OOOOIIII is easier to understand than OPOOOOIIII. **Note**: The colon is required in GR command and is not neutral. (Unit Ratio). Example: **GR4:1**

Some IDeal™ commands request a response from the control.Responses will always be preceded by an asterisk (*) which notifies the other controls on a daisy chain to ignore the subsequent response characters preceding the next command delimiter. For example,the Input State (IS) command might return *AF09<cr> Your computer program will need to mask the asterisk before decoding the value returned.

You can document your programs by placing comments between brackets. For example:{this is a comment}. To maximize program storage space,the control "strips off" these comments when a program file is downloaded.

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>AH</td><td>Anti Hunt
0 = Anti-Hunt Disabled default
1 = Anti-Hunt Enabled
Not recommended to change unless using tuning software. See tuning software help file for further info</td><td><n>AHi,i</td></tr>
<tr><td>AM</td><td>Acceleration Maximum</td><td><n>AMr,r</td></tr>
<tr><td>AU</td><td>Acceleration Units
i=0 units/s² (where “units” is a string defined by the DU command)
1 rps²
2 seconds
Example: AU0 (unit/s² on axis one)</td><td><n>AUi,i</td></tr>
<tr><td>AW</td><td>Anti-Inertial Windup
0 = Anti-Inertial Windup Disabled, (default)
1 = Anti-Inertial Windup Enabled
Not recommended to change unless using tuning software. See tuning software help file for further info</td><td><n>AHi,i</td></tr>
<tr><td>BV</td><td>Clamp Break Velocity</td><td><n>BVr,r</td></tr>
<tr><td>CU</td><td>Clamp Current Units
i = 0 Amps
i = 1 % max Torque</td><td><n>CUi,i</td></tr>
<tr><td>DF</td><td>Display Format
Configures the 4 keypad run time display quadrants. DF takes 4 parameters where i is an integer representing a display data type per quadrant.
User defined text is limited to 10 characters per field.

<table>
<tr><td>i = 0</td><td>Blank</td><td>i = 8</td><td>CUR2</td></tr>
<tr><td>i = 1</td><td>POS1</td><td>i = 9</td><td>INPUTS</td></tr>
<tr><td>i = 2</td><td>POS2</td><td>i = 10</td><td>OUTPUTS</td></tr>
<tr><td>i = 3</td><td>POS1 + UNIT</td><td>i = 11</td><td>OPTOS</td></tr>
<tr><td>i = 4</td><td>POS2 + UNIT</td><td>i = 12</td><td>SA_STATUS1</td></tr>
<tr><td>i = 5</td><td>VEL1</td><td>i = 13</td><td>SA_STATUS2</td></tr>
<tr><td>i = 6</td><td>VEL2</td><td>I = 14</td><td>SS_STATUS</td></tr>
<tr><td>i = 7</td><td>CUR1</td><td>“ “</td><td>User defined text in quotes</td></tr>
</table></td><td><n>DFi,i
<n>DF” Text”,i,i,i</td></tr>
<tr><td>DU</td><td>Distance Unit Label
i = 0 steps (fixes GR @ 1:1)
<table>
<tr><td>1 = rev</td><td>5 = cm</td><td>9 = deg</td><td>13 = arcmin</td></tr>
<tr><td>2 = inch</td><td>6 = mm</td><td>10 = radian</td><td>14 = %</td></tr>
<tr><td>3 = mil</td><td>7 = yard</td><td>11 = grad</td><td>15 = index</td></tr>
<tr><td>4 = meter</td><td>8 = foot</td><td>12 = arcsec</td><td>16 = µm</td></tr>
</table></td><td><n>DUi,i</td></tr>
</table>

Serial Setup Commands These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.		
Command	**Command Description and Application Examples**	**Syntax**
DY	**Scan Delay** Where i is the number of ms Example: DY500 (500ms) Default is 100ms	**<n>DYi**
EL	**Enable Line Polarity** Fixed in SmartDrive	**<n>EL1,1**
EM	**Encoder Mode** 3 = Servo Closed Loop Example: EM3,3 (servo closed loop) B8961/2 Fixed encoder mode of servo closed loop.	**<n>EMi,i**
ER	**Encoder Resolution** Where i is an even integer Example: ER8000 (8000 counts/rev) B8961/2 Fixed encoder resolution based on motor type. (see MT command)	**<n>ER,i**
ET	**End of Travel Switch Polarity** Selects the polarity of EOT (End of Travel) switches i = 0 NORM OPEN i = 1 NORM CLOSED	**<n>ETi,i**
FA	**Accel Feed-Forward Gain** 0 = Anti-Inertial Windup Disabled, (default) 1 = Anti-Inertial Windup Enabled	**<n>FAn,n**
FE	**Following Error Limit** Example: FE1000 (axis one 1000 counts)	**<n>FEi,i**
FL	**Fault Line Polarity** Fixed in SmartDrive	**<n>FL0,0**
FV	**Velocity Feed-Forward Gain** 0 = Anti-Inertial Windup Disabled, (default) 1 = Anti-Inertial Windup Enabled	**<n>FVn,n**
GR	**Units Ratio** Example: GR4:1 (4 motor revolutions per distance (DU) unit)	**<n>GRi:i, i:i**
HE	**Home Edge** 0 = Negative Direction, 1 = Positive Direction Example: HE1 (one positive)	**<n>HEi,i**
HF	**Home Final Direction** 0 = Negative Direction, 1 = Postive Direction Example: HF1 (one positive)	**<n>HFi,i**
HM	**Homing Mode** i = 0 Switch only i = 1 Switch then Z Channel i = 2 Z Channel only	**<n>HM**

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.

Command	Command Description and Application Examples	Syntax
HO	**Home Offset** Example: HO1.0 (axis one 1.0 distance unit)	**<n>HO±r,** **±r**
HS	**Home Switch** 0 = Normally Closed, 1 = Normally Open Example: HS1 (uses a normally open home switch - this is the default setting.	**<n>HSi,i**
ID	**Input Definition** Example: ID: UUUUUUUU The first 8 inputs are unassigned. All 8 input states must be specified. See also: OD, OP Note: The G (Registration) Command is only valid for Input 1.	**<n>IDaaa** aaaaaaaa aaaaa
IL	**Idle Mode** Not used in B8961/2 fixed at 0.	**<n>IL0,0**
IR	**In-Range Window (B896X only)** Sets In-Range window in motor steps Default: 0 Not recommended to change unless using tuning software.	**<n>IRi,i**
IT	**In-Range Time (B896X only)** Sets In-Range time limit within In-Range Window in milliseconds. Default: -10 Not recommended to change unless using tuning software.	**<n>ITi,i**
JA	**Jog Acceleration** Example: JA.01 (.01-Units selected by AU command.)	**<n>JAr,r**
JE	**Jog Enable** 0 = Jog Disabled, 1 = Jog Enabled Example: JE1 (enabled)	**<n>JEi,i**
JH	**Jog High Velocity** Example: JH5.0 (5 in units selected by VU command.)	**<n>JHr,r**
JL	**Jog Low Velocity** Example: JL1.5 (1.5 in units selected by VU command.)	**<n>JLr,r**
KI	**Integral Gain Factor** Not recommended to change unless using tuning software. See tuning software help file for further info.	**<n>KIr,r**
KP	**Position Gain Factor** Not recommended to change unless using tuning software. See tuning software help file for further info.	**<n>KPr,r**
KV	**Velocity Gain Factor** Not recommended to change unless using tuning software. See tuning software help file for further info.	**<n>KVr,r**

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.

Command	Command Description and Application Examples	Syntax
MD	**Motor Direction Reference** 0 = Positive direction, 1 = Negative direction Example: MD0 (one positive)	**<n>MDi,i**
MH	**Motor Inductance** Fixed in B8961/2 by motor type (see MT)	**<n>MHi,i**
MI	**Motor Current** Fixed in B8961/2 by motor type (see MT)	**<n>MIi,i**
MR	**Motor Resolution** Default: 8000 steps/rev Fixed in B8961/2 by motor type (see MT)	**<n>MRi,i**
MT	**Motor Type** Selects a built-in motor type. Note: Linear motor types are negative i = 0 NONE **1** B23: 110V **2** B23: 220V **3** B32: 110V **4** B32: 220V **5** B41: 110V **6** B23: 220V **7** B23: 110V **8** B23: 110V **9** OTHER **10** B12: 110V **11** B22: 110V **12** B22: 110V **13** B23H: 110V **14** B23H: 110V **15** B31: 110V **16** B31: 110V **17** B33: 110V **18** B33: 110V **19** B40: 110V **20** B40: 110V **21** B42: 110V **22** B42: 220V **23** BN21: 110V **24** BN23: 110V **25** B31: 110V **26** B32: 110V **-1** 2504: 110V 10µ **-1** 2504: 220V 10µ **-1** 2504: 110V 5µ **-1** 2504: 220V 5µ **-1** 2504: 110V 1µ **-1** 2504: 220V 1µ **-1** 2504: 110V 0.5µ **-1** 2504: 220V 0.5µ **-1** 2506: 110V 10µ **-10** 2506: 220V 10µ **-11** 2506: 110V 5µ **-12** 2506: 220V 5µ **-13** 2506: 110V 1µ **-14** 2506: 220V 1µ **-15** 2506: 110V 0.5µ **-16** 2506: 220V 0.5µ **-17** 2508: 110V 10µ **-18** 2508: 220V 10µ **-19** 2508: 110V 5µ **-20** 2508: 220V 5µ **-21** 2508: 110V 1µ **-22** 2508: 220V 1µ **-23** 2508: 110V 0.5µ **-24** 2508: 220V 0.5µ **-25** 3804: 110V 10µ **-26** 3804: 220V 10µ **-27** 3804: 110V 5µ **-28** 3804: 220V 5µ **-29** 3804: 110V 1µ **-30** 3804: 220V 1µ **-31** 3804: 110V 0.5µ **-32** 3804: 220V 0.5µ **-33** 3806: 110V 10µ **-34** 3806: 220V 10µ **-35** 3806: 110V 5µ **-36** 3806: 220V 5µ **-37** 3806: 110V 1µ **-38** 3806: 220V 1µ **-39** 3806: 110V 0.5µ **-40** 3806: 220V 0.5µ **-41** 3808: 110V 10µ **-42** 3808: 220V 10µ **-43** 3808: 110V 5µ **-44** 3808: 220V 5µ **-45** 3808: 110V 1µ **-46** 3808: 220V 1µ **-47** 3808: 110V 0.5µ **-48** 3808: 220V 0.5µ **-49** 3810: 110V 10µ **-50** 3810: 220V 10µ **-51** 3810: 110V 5µ **-52** 3810: 220V 5µ **-53** 3810: 110V 1µ **-54** 3810: 220V 1µ **-55** 3810: 110V 0.5µ **-56** 3810: 220V 0.5µ **-57** OTHER	**<n>MT**
MV	**Maximum Velocity** Example: MV50.0 (axis one 50 units selected by VU command)	**<n>MVr,r**
OD	**Output Definition** Example: OD:PPPPPPPP All 8 outputs defined as Programmable outputs. All 8 output states must be specified. See also ID, OP	**<n>ODaa** aaaaaaaa aaaaaa

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.

Command	Command Description and Application Examples	Syntax
OE	**Ouput States on Event** Configures output states on an event specified by a. OPTO positions 9-16 are only definable if configured as an ouput using OP command. a = P (Power-Up) F (Fault) S (Stop/Kill) i = 0 Off 1 on X No Change Example: OEF, 0001XX01XXXXXXXX (Output state on fault turn 1,2,3, 7 off; 4,8 on; all else leave unchanged)	**<n>OEa,iii** iiiiiiiiiiii
OP	**OPTO Configuration** i = Input, O = Output Example: OP:IIIIOOOO First four configured as inputs, last four as outputs	**<n>OPIIII OOOO**
PG	**Position Maintenance Gain** Sets position maintenance correction gain. i = 1 to 32,767 Not used by B8961/2. Fixed at 0.	**<n>PGi,i**
PU	**Power-Up Program** Example: PU105 (Runs program number 105 on power-up.)	**<n>PUi**
PV	**Postion Maintenance Max. Velocity** Sets position maintenance maximum correction velocity (Units specified by VU command). Not used by B8961/2. Fixed at 0.	**<n>PVi,i**
PW	**Password** Up to 4 characters: a-a, A-Z, 0-9 Entering a dash (-) will clear the password. Skipping a parameter will leave the password unchanged (see examples below) <u>Examples using PW:</u> Example 1: **PW4FT,Q12h** Set the OPRATR password to 4FT and the ADMIN password to Q12h Example 2: **PW,New** Changes ADMIN password to New; leaves OPRTATR password unchanged Example 3: **PW-,-** Clears both the OPRATR and ADMIN passwords	**<n>PWaa aa,aaaa**
RE	**Rest Mode** Not used by B8961/2. Fixed at 0.	**<n>RE0,0**

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartDrive according to the choice made in the SETUP dialog boxes.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>SN</td><td>Scan Conditions
Conditions stopping input program select line (BIN/BCD) scanning are represented by: i”
<table><tr><td></td><td>ESC</td><td>Stop</td><td>Limit+</td><td>Limit-</td><td>Kill</td><td>Fault</td><td>Interrupt</td></tr><tr><td>SN</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td></tr></table>0 = Continue program select scanning,
1 = Stop program select scanning on this condition
Example: SN:0111111 (all input conditions, except pressing the ESC key stop program select line scanning)
See also: ID</td><td><n>SNiiiiii
ii</td></tr>
<tr><td>SR</td><td>Stop Deceleration Rate
Example: SR100 (100 rps2)
Note: Stop Deceleration Rate units are always in rps2 and are not user selected.</td><td><n>SRr,r</td></tr>
<tr><td>UN</td><td>Unit Number
Example: UN5 (sets unit address to 5)</td><td><n>UNi</td></tr>
<tr><td>VU</td><td>Velocity Units
<table><tr><td>0</td><td>units/sec (where units is a string defined by the DU command)</td></tr><tr><td>1</td><td>units/sec (where units is a string defined by the DU command)</td></tr><tr><td>2</td><td>rps (motor revs/sec)</td></tr><tr><td>3</td><td>rpm (motor revs/min)</td></tr></table>Example: VU2 (axis one set to rps)
See also: GR, DU</td><td><n>VUi,i</td></tr>
</table>

Serial Programming Commands

The commands in this category may be sent to the SmartDrive's buffer and executed on a first-in first-out (FIFO) basis. This execution does not require you to create or download any program to the SmartDrive's volatile memory. See *Chapter 6, Programming Commands for complete definitions.*

Command	Command Description and Application Examples	Syntax
AC	**Acceleration** Example: AC10	**ACr,r**
DA	**Distance Absolute** Example: DA15	**DAr,r**
DC	**Distance to a Change** Example: DC20	**DC+r,+r**
DE	**Deceleration** Example: DE.2	**DE+r,+r**
DI	**Distance Incremental** Example: DI 3.42	**DI+r,+r**
EA	**Enable Axis** i=0 (disable drive), i=1 (enable drive), i=2 (standby) Example: EA0 disables axis	**EAi,i**
GH	**Go Home** Example: GH10	**GH+r**
GI	**Go Immediate**	**GI, GIi,i**
GO	**Begin Move**	**GO, GOi,i**
MC	**Move Continuous**	**MC+,+**
OT	**Set Outputs** Example: OT5,101 (turns on outputs 5 and 7, turns off output 6)	**OTi,iii...** **OTiii...**
" "	**Send String Over Serial Port** Example: GO "End of Move" (sends "End of Move" out serial port after move) See also: (TERM)=	**"user text"**
RG	**Registration** Example: RG3.5	**RGr,r**
SP	**Set Position** Example: SP15.0 (set axis position to 15.0)	**SPr,r**
SQ	**Square Root** Example: SQ16, (x)	**SQr,(var)**

Serial Programming Commands

The commands in this category may be sent to the SmartDrive's buffer and executed on a first-in first-out (FIFO) basis. This execution does not require you to create or download any program to the SmartDrive's volatile memory. See *Chapter 6, Programming Commands* for complete definitions.

Command	Command Description and Application Examples	Syntax
ST	**Stop on Input** Example: ST4 DI50 GO (motor will decelerate to a stop or remain stopped if input 4 is activated). ST0 disables the input	**STi,i**
TD	**Time Delay** Example: OT1,1 TD.12 OT1,0 Turn Output 1 on for 120 msec.	**TDr**
(TERM)=	**Output Variable with Serial Port** Similar to MS command (used with front panel). Outputs variable the serial port Built-in variables or user defined variables can be output. Can be used with Example: (TERM)=(POS1) See also " " (Send String Over Serial Port)	
VE	**Velocity** Example: VE50 (sets speed for the next move to 50, in units set by the VU command)	**VEr,r**
WT	**Wait on Input Condition** Example: WT1,0 GO (wait for input 1 to turn off before starting move) #1 = axis to "wait on"	**WTi,ii...** **nWTiii...** **<n>WT expression** **<n>WT#i**

Commands Not Available In Hosted Mode The following set of commands can only be executed if they are part of, or within, a program:	
Command	**Command Name**
BR	Break
EB	End Block
GS	Go Sub
GT	Go To
IF	If
IV	Input Variable
LP	Loop
MS	Display Message
EN	End Routine
FK	Function Key
LU	Loop Until
LW	Loop While
WT	Wait

Serial Immediate Status Commands

Serial Immediate Status commands are processed immediately upon receipt, rather than waiting in the buffer for previous commands to finish.They can be issued while a program is running,or while motion is in progress.They can't be stored within a program.

Using Immediate Status Commands

Serial Immediate Status commands are provided for two purposes.One is to allow a host control to query the SmartDrive,in real time, for system,position,and I/O status.The second is to provide a means to do in-depth troubleshooting via RS-232C.These commands will interrupt the SmartDrive and generate a return.They do not affect operation of the SmartDrive.

In a typical hosted-mode application,all machine operations and decisions are performed by a high level device.Motion commands are generated,and downloaded to the SmartDrive by this host device,such as a computer or PLC.The following commands are provided so that the host can verify the status of the SmartDrive before commanding motion.The System Status (SS) command returns overall system information,and indicates general faults.The Axis and Drive (SAi and SDi) commands can then be used to provide more detailed,axis specific information.

These commands are also an invaluable system troubleshooting aid.Since they are immediate commands,they will generate a response from the SmartDrive even if it is in the middle of move, waiting for an input condition to become true,etc. Checking the System Status,and the I/O Status will give you enough information to

explain what the SmartDrive is doing.If a fault is indicated,the Drive and Axis Status commands can give detailed,axis specific information.

Summary of Immediate Status Commands

Command	Syntax	Purpose
Clear Command Buffer	<n>CB	Clears the terminal input buffer and buffered command buffer
Input States	<n>IS	Real-time status of discrete and OPTO inputs
Kill	<n>K	Issues immediate halt to current and programmed motion
Model Number	<n>MN	Returns unit model number over RS-232C
Output States	<n>OS	Real-time status of discrete and OPTO outputs
Current Position	<n>PA1, <n>PA2	Real-time position, in user units, of axis 1, axis 2
Stop	<n>S	Issues program termination, decelerates to a halt
Axis Status	<n>SA1, <n>SA2	Returns axis specific status (i.e. limit and home states) of axis 1, axis 2
Drive Status	<n>SD1, <n>SD2	Returns drive specific status (i.e. type of amp fault) of axis 1, axis 2
System Status	<n>SS	Returns general system status and operation

Serial Immediate Status Commands

Note: All but the S and K commands require an address.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>CB</td><td>Clear Command Buffer
Clears the terminal input buffer and buffered command buffer</td><td><n>CB</td></tr>
<tr><td>IS</td><td>Tell Input States
Returns the current state (on or off) of the 8 inputs. The status is returned as a four digit hexadecimal number, preceded by an asterisk. The least significant digit represents the binary value of inputs 4-1.

Example: IS returns *00F6<cr> with the input conditions shown in this table

<table>
<tr><td colspan="8">n/a</td><td colspan="8">inputs</td></tr>
<tr><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>on</td><td>on</td><td>on</td><td>off</td><td>on</td><td>on</td><td>off</td></tr>
<tr><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">F</td><td colspan="2">6</td></tr>
</table>

Your computer program will have to decode the hexadecimal number to determine the state of any each input.</td><td><n>IS</td></tr>
<tr><td>K</td><td>Kill
Issuing the K command causes the control to abruptly stop commanding further motion and temrinates program execution. No deceleration ramp is used with this command. Caution should be used in issuing this command because of the damage instantaneous deceleration could cause to mechanics. The Stop commands provides a more controlled halt.</td><td><n>K</td></tr>
<tr><td>MN</td><td>Model Number
Returns the unit model number..
MN command responses are: *S6961 *S6962 *B8961 or B8962</td><td><n>MN</td></tr>
</table>

Serial Immediate Status Commands

Note: All but the S and K commands require an address.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>OS</td><td>Tell Output States
Returns the current state (on or off) of the 8 inputs. The status is returned as a four digit hexadecimal number, preceded by an asterisk. The least significant digit represents the binary value of inputs 4-1.

Example: IS returns *00F6<cr> with the input conditions shown in this table.

<table>
<tr><td colspan="8">n/a</td><td colspan="8">inputs</td></tr>
<tr><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>on</td><td>on</td><td>on</td><td>off</td><td>on</td><td>on</td><td>off</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">F</td><td colspan="4">6</td></tr>
</table>

Your computer program will have to decode the hexadecimal number to determine the state of any each input.</td><td><n>IS</td></tr>
<tr><td>PA</td><td>Tell Absolute Position
Reports current position in user units based on encoder mode selected. Can report specifically commanded or encoder position when PAa,n is used.
a = Commanded Position
E (Encoder Position)
i = 1-2 axis
Examples: PA2 Returns axis #2 position
PAC,1 Returns axis #1 commanded position
Since B8961/2 is always servo-closed loop, no difference between commanded and encoder position</td><td><n>PAn
<n>PAa,i</td></tr>
<tr><td>RS</td><td>Reset System
Re-initializes, or warm boots, the control software to its power-up state. The initialization process takes about 10 seconds to complete. Programs and configuration settings are not erased. This command is the equivalent of cycling power.</td><td><n>RS</td></tr>
<tr><td>S</td><td>Stop
Terminates program execution and immediately decelerates each motor to a halt (at a rate set by the SR command). Functions the same as pressing the ESC key on the IDC keypad or activating an input defined as a Stop input.</td><td><n>S</td></tr>
</table>

Serial Immediate Status Commands

Note: All but the S and K commands require an address.

Command	Command Description and Application Examples	Syntax
SA	**Tell Axis Status** Returns the current axis status as a four digit hexadecimal number, preceded by an asterisk. Your controller program will decode the hexadecimal number to determine the axis status. Example: SA1 returns *002A<cr>. This means Axis 1 is not moving, the last move completed successfully, and the home switch is on.	**<n>SA1**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	on	off	on	off	on	off
0				0				2				A			

Description	bit #	Parameter Definition
Not Moving /Moving	1	1=Steps being sent to the amplifier 0=No steps being sent
At Velocity	2	1=Stepping at a constant rate (includes zero velocity) 0=Step rate is changing
In Range	3	B8961/2 only
Move Command Complete (Same as Move Done Output)	4	1=The correct number of steps were sent without an amp fault,following error, or hitting an End of Travel limit. 0=Reset to zero at the beginning of each move.
Home Successful	5	1=The last homing move was successful 0=At power up,reset to zero at the start of the next jog,GO,or GH.
Home Switch Status	6	Hardware status of home switch. 0=off,1=on
-Limit Switch Status	7	Hardware status of limit switch 0=off,1=on,limits require a NC switch
+Limit Switch Status	8	Hardware status of limit switch 0=off,1=on,limits require a NC switch
-Limit Switch Latched	9	1=Set when a move is terminated by a limit in the - direction.Cleared at the start of a move in the + direction. 0=At power up or reset,even if on the - limit.
+Limit Switch Latched	10	1=Set when a move is terminated by a limit in the + direction.Cleared at the start of a move in the - direction. 0=At power up or reset,even if on the + limit
RESERVED	11	State undefined,should be masked
RESERVED	12	State undefined,should be masked
RESERVED	13	State undefined,should be masked
RESERVED	14	State undefined,should be masked
RESERVED	15	State undefined,should be masked
RESERVED	16	State undefined,should be masked

Serial Immediate Status Commands

Note: All but the S and K commands require an address.

Command	Command Description and Application Examples	Syntax
SD	**Tell Drive Status** Returns the current drive status as a four digit hexadecimal number, preceded by an asterisk. Your controller program decodes the hexadecimal number to determine the drive status.	**<n>SD1**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	off	on	off	off	off	off
0				0				1				0			

Description	bit #	Parameter Definition
Following Error	1	1=Following error occurred 0=At power up and reset. Set to zero at the start of the next move
Over-current	2	1=Over Current (requires reset to clear) 0=At power up and after reset
Thermal Fault	3	1=Thermal fault in the motor or drive (requires reset to clear) 0=At power up and after reset
RMS Over-current B8961/2 only	4	1=RMS current limit exceeded (requires reset to clear) 0=At power up and after reset
Drive Enabled	5	1=Enable Drive (see also EA1) 0=Disable Drive (see also EA0)
RESERVED	6	State undefined,should be masked
RESERVED	7	State undefined,should be masked
Torque/Position B8961/2 only	8	1=Amplifier in torque mode (TM1) 0=Amplifier in position mode (factory default)
Amplifier Fault	9	1=The amplifier is faulted.Requires a power cycle to reset. 0=At power up or reset
RESERVED	10	State undefined,should be masked
RESERVED	11	State undefined,should be masked
RESERVED	12	State undefined,should be masked
RESERVED	13	State undefined,should be masked
RESERVED	14	State undefined,should be masked
RESERVED	15	State undefined,should be masked
RESERVED	16	State undefined,should be masked

Serial Immediate Status Commands

Note: All but the S and K commands require an address.

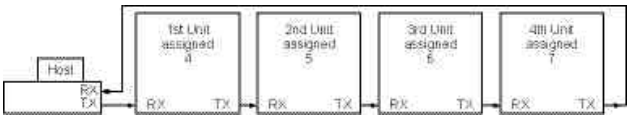
Command	Command Description and Application Examples	Syntax
SS	**Tell System Status** Returns the current system status as a four-digit hexadecimal number, preceded by an asterisk. Your controller program decodes the hexadecimal number to determine the system status. Example: SS returns *0001<cr> means there are no amplifier faults, and no programs running - SmartDrive is ready to process any buffered RS-232C command.	**<n>SS**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	off	on	off	off	off	on
0				0				0				1			

Description	bit #	Parameter Definition
Ready to Buffer RS-232C Commands	1	1=Ready to buffer RS-232C commands 0=Initializing from a power-up or reset,or unchecked errors exist.Any buffered commands sent will be discarded.
FLASH Error	2	1=Non-volatile memory checksum error, all programs were deleted on power up. 0=Non-volatile memory checksum OK
Program Running	3	1=Running a program 0=Not running a pre-defined program
FK Active	4	1=Paused waiting for a function key. 0=Not waiting at a FK command
WT Active	5	1=Paused waiting for a input condition. 0=Not waiting at a WT command
TD Active	6	1=Paused at a time delay 0=Not waiting at a TD command
Waiting for IV	7	1=Paused,waiting a variable input 0=Not waiting at a IV command
Buffer Full	8	1=RS-232C buffer 75% full Total Capacity: 2k 0=RS-232C buffer less than 60% full
Axis #1 Fault	9	1=Amp fault,following error, move stopped by limit switch (see SAi and SDi for more detailed fault information) 0=No faults
Axis #2 Fault	10	1=Amp fault,following error, move stopped by limit switch (see SAi and SDi for more detailed fault information) 0=No faults
RESERVED	11	State undefined,should be masked
Program Select Scanning	12	1=BCD and Binary program select scanning enabled. 0=A Stop Scan condition has occurred or no inputs are configured as program select lines.
Data Download Status	13	1=Data Transfer failed (program memory overflow) 0=Data successfully received
RESERVED	14-16	State undefined,should be masked

Serial Supervisory Commands

The following commands control program uploading, downloading, deleting, execution, etc. All of these commands are fully defined in this section.

Command	Command Description and Application Examples	Syntax
AA	**Auto Address** The AA command automatically addresses SmartDrive units in a daisy chain. It assigns an address to each unit on the daisy chain. This allows the units to be wired in a daisy chain without setting each unit's address manually. The AA command parameter n indicates the value in which the addressing sequence will begin.  In the example above, the Host issues an AA4 and the units are addressed 4, 5, 6, 7. This offers the convenience of adding a new unit anywhere in the daisy chain without manually re-addressing all the other units. Just connect the new unit, issue an AA command from the new unit with the address of the new unit as the AA parameter, i.e. AAi.	**<n>AA or AAi**
DP	**Delete Program** Erases a program from memory, where i is the program number. This is equivalent to pressing the delete key on the keypad and entering the program number. Range: 1-199 (1-400 with 30K memory option) Example: DP99 (deletes program number 99).	**<n>DPi**
DR	**Download Program to RAM** Begins downloading a program from the host to the control's RAM, rather than non-volatile memory. (Also see the PR command description) These programs will be lost after a reset or power cycle. The program string must end in EP. The commands between DR and EP do not need a device address. The DR command is typically used when the control is operated exclusively via a host controller which constantly downloads and executes programs. This increases the usable life of the FLASH. Range: 1-199 (1-400 with 30K memory option) Example: 1DR50 AC4 DE4 VE30 LP6 DI10.5 GO EN 1EP RN50 Downloads program #50 to Unit #1's RAM, then runs program #50	**<n>DRi**

Serial Supervisory Commands

The following commands control program uploading, downloading, deleting, execution, etc. All of these commands are fully defined in this section.

Command	Command Description and Application Examples	Syntax
EC	**RS-232C Echo Enable/Disable** 0 = echo Disabled, 1 = echo Enabled. Example: EC0 (echo off) The RS-232C Echo must be enabled for daisy chain operation	**<n>ECi**
EP	**End Program Definition** Denotes the end of a program definition. All program definitions must begin with nPRi or nDRi and end with EP. Example: PR15 [part A] AC4 VE30 DI10.5 GO EP	**<n>EP**
EX	**Ends Upload All or Load All** Signals the end of a upload all (UA) or load all (LA) sequence. EX is sent by the SmartDrive to the host after completing a UA. EX is sent by the host to the SmartDrive to terminate a LA.	**<n>EX**
LA	**Load All** Sent to the SmartDrive before downloading a long list of setup parameters and programs. This command will disable the non-addressed units so that each setup parameter doesn't need an address. Must be followed by an EX to reestablish the daisy chain communications.	**<n>LA**
LS	**List Programs** Lists number of programs, memory usage, and the current available memory of the SmartDrive. Just like Edit/List from the keypad.	**<n>LS**
OC	**Original Configuration** Returns the FLASH to its original factory-default state. The command buffer is cleared, all programs are erased, and all configuration settings are returned to their default values Can be done with keypad, pressing sequentially. . → DEL (decimal, right arrow, delete)	**<n>OC**
PR	**Define Program** Starts a program definition. Just like the DR command, but writes the SmartDrives non-volatile EEP ROM memory. Example: PR25 AC.1 VE5 DI10 GO EP (uses only a program number). Example: PR25 [P/N 170-001] AC.1 VE5 DI10 GO EP (uses optional program name).	**<n>PRi**

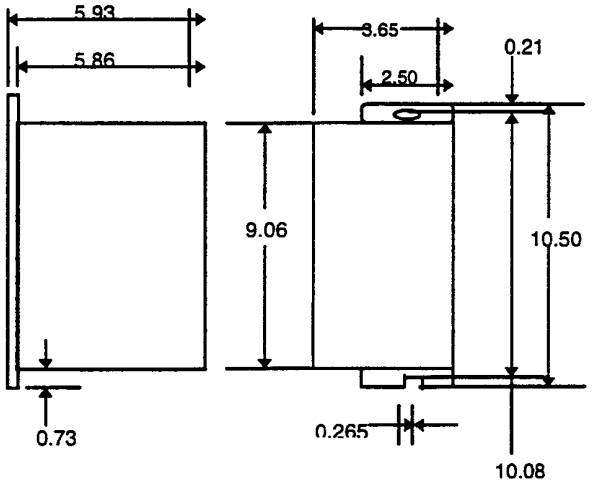
Serial Supervisory Commands The following commands control program uploading, downloading, deleting, execution, etc. All of these commands are fully defined in this section.		
Command	**Command Description and Application Examples**	**Syntax**
RN	**Run Program** This commands any program, by number only. The RN command does not support the optional program names. Example: RN25	**<n>RNi**
SW	**Software/Firmware Version** SW returns firmware version SW1 returns DSP, FPGA, and firmware versions	**<n>SW**
UA	**Upload All** Uploads all setup parameters and programs from unit n. SmartDrive sends an EX to terminate upload.	**<n>UA**
UL	**Upload Program** Range: i=1-199 (program #) (up to 400 programs with 30K option) Uploads program number i to the host - SmartDrive adds brackets. Example: 1UL2 Uploads program 2 from unit #1. Response: { [part A] AC4 VE30 DI10.5 GO }	**<n>ULi**

Hardware Reference

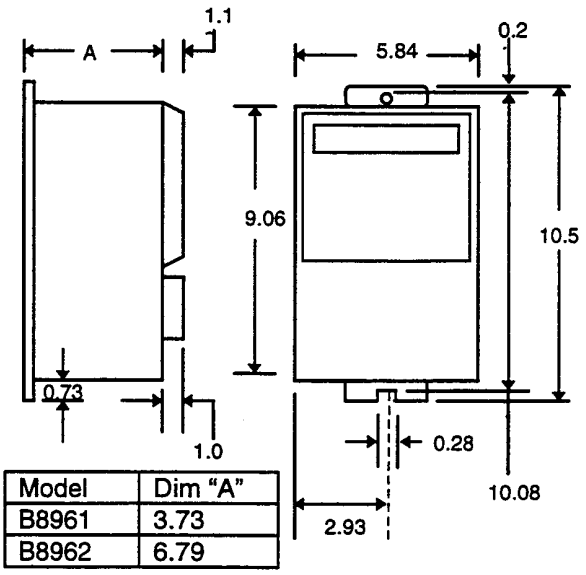
B8961 and B8962 Smart Drive Mounting

Your B8000 Series Smart Drive arrived ready for a minimum depth cabinet mounting.By moving the mounting bracket to the narrow side of the heatsink, you can mount the Smart Drive in a minimum width configuration.

MINIMUM WIDTH MOUNTING

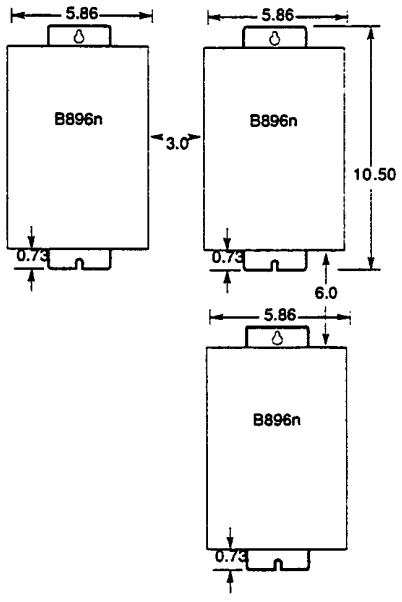


MINIMUM DEPTH MOUNTING



Model	Dim "A"
B8961	3.73
B8962	6.79

The B8961 and B8962 depend on air flow for cooling.In all but the least demanding applications,the following mounting guidelines should be observed:



- Allow clearance (approximately one inch) for a -FK1 or FK2 fan kit if the available airflow appears marginal.The fan kit adds 1.05 in to the depth of the B896n if mounted in the Minimum Depth Configuration.It adds 1.05 in to the width if mounted in the Minimum Width Configuration.
- The heat sink fins should be oriented vertically for convection currents to flow along the fins.
- The vertical clearance between a B8000 drive and other equipment or surfaces of the enclosure should be a minimum of 6.0 inches,and horizontal clearance should be a minimum of 3.0 inches.
- High heat-generating equipment should not be mounted directly below the B8000 Smart Drives.

Forced air cooling may be necessary if air flow is extremely poor, or the ambient temperature is excessive.

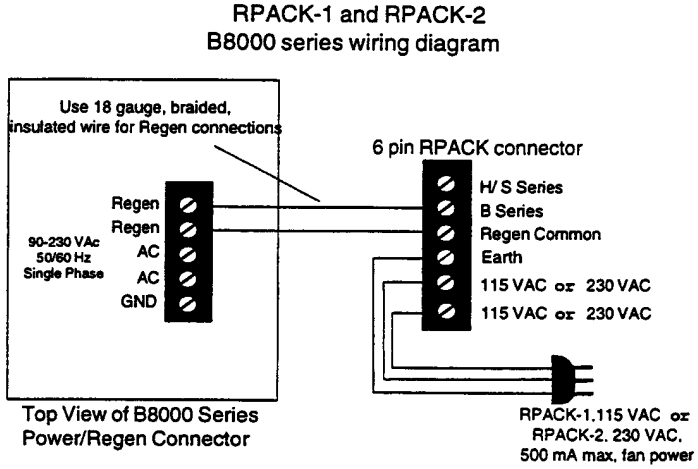
FK1 and FK2 Fan Kit

The FK1 and FK2 Fan Kit Option may be necessary if air flow across the B896n heat sink is extremely poor, or if the ambient temperature is excessive.The FK1 and the FK2 are the 110VAC and 220VAC versions of IDC's fan kit.They can be ordered individually for drive mounting in the field,or ordered as part of a system (i.e.B8961-FK2) for factory mounting.Since the B896n supplies power to the fan no additional wiring or mounting is required.

See Appendix C for additional information on the fan kit,including mounting dimensions.

RPACK-1 and RPACK-2 Mounting Installation

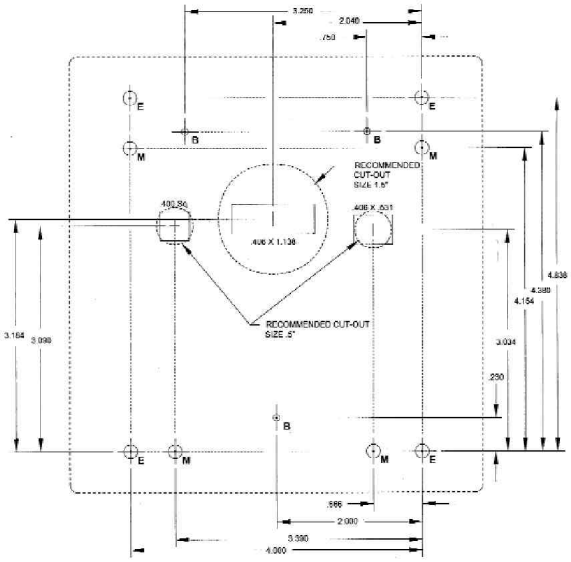
The RPACK-1 and the RPACK-2 are packaged power dump devices consisting of high voltage power resistors, a heatsink and a cooling fan. They are accessories to IDC's B Series controls.The RPACK-1 has a 110VAC fan while the RPACK-2 has a 220VAC fan. Please note that unlike a fan kit, the RPACK is a standalone box that does not share a power supply with the B8001.



An RPACK is typically used when it is necessary to very rapidly decelerate heavy vertical loads or large inertial loads.The RPACK-1 dissipates the kinetic and potential energy that would otherwise cause the drive to shutdown due to an Over-Voltage or Regen fault.Using an RPACK-1 can help reduce overall cycle time by allowing a higher deceleration rate than would be possible without it.

See Appendix D for additional information on the RPACK,including mounting dimensions.The recommended panel spacing is also the same.Electrical connections should be made according to the diagram above.

Remote Keypad Mounting:



B8000 Smart Drive Hardware Specifications

General

Command Interface:

Type:RS-232C serial type,3 wire implementation (TX,RX & COM)
Parameters:9600 Baud,8 data bits,1 stop bit,no parity
Configuration:Up to 99 units can be supported over RS-232C on a daisy chain

Inputs:

Programmable inputs:8 optically isolated programmable inputs,Limits,Home,
All inputs are active low, optically isolated inputs.Min current:12 mA,Max:35 mA
250 mA max.current available from on board 12Vdc supply.

Encoder:Differential quadrature incremental (TTL).Max freq.of 1.5 MHz.
Power +5VDC @ 200 mA output power for each external encoder

Outputs:

Programmable Outputs:8 optically isolated,open collector, programmable outputs,
Isolated Pull Up:5-24Vdc,100 mA max.current per output.
250 mA max.current available from 12Vdc supply.

Diagnostic Monitor port:.5-4.5Vdc analog,centered at 2.5V.Variables of vel,pos/vel error

OPTO compatible I/O: 8 OPTO compatible:Any G4 I/O module; analog & temperature

Controller Performance:

Positioning Range: +/- 2,147,483,647 steps; absolute and incremental.
Velocity Range: 0 to 1,250,000 steps/sec (motor and voltage dependent)
Acceleration Range: 0.01 to 999.99 rev/sec^2

Environmental:

Drive Temperature:Thermal Shutdown occurs if the drive heatsink temp exceeds 55˚C
Motor Temperature: Max winding temp 135˚C (thermostat protected)
Humidity: 0-90% non-condensing

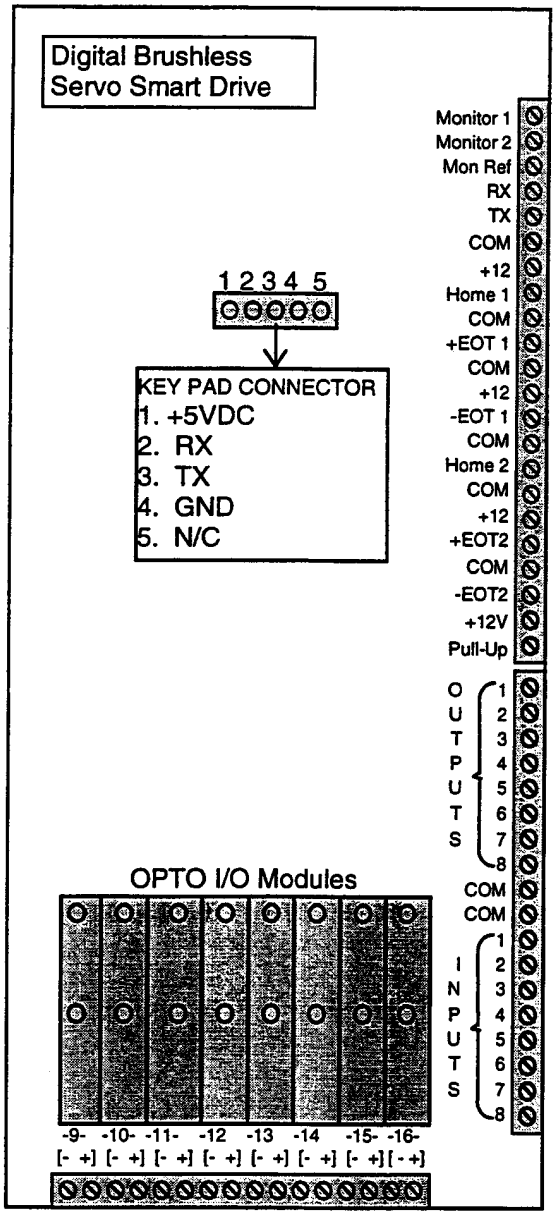
Power Input:

90-230VAC, single phase, 50/60Hz
1150VA @ 115VAC,2300VA @ 230VAC for the B8961
2300VA @ 115VAC,4600VA @ 230VAC for the B8962

Motor Output

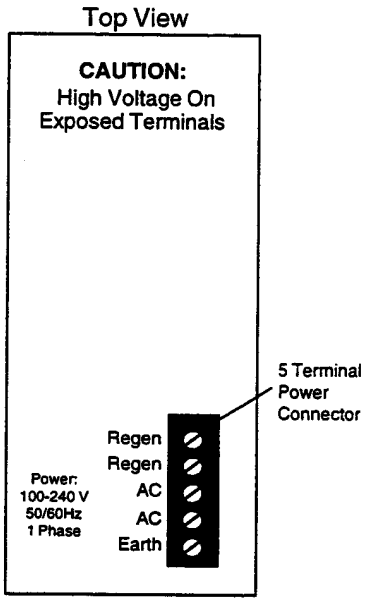
Current: 5 amps continuous, 10 amps peak
Protection:Protected against phase to phase shorts,shorts to ground.Internally fused.

B8961 and B8962 Hardware Connections

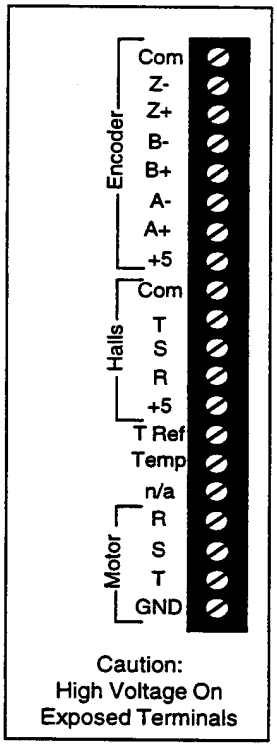


B8961 and B8962 I/O Pinouts

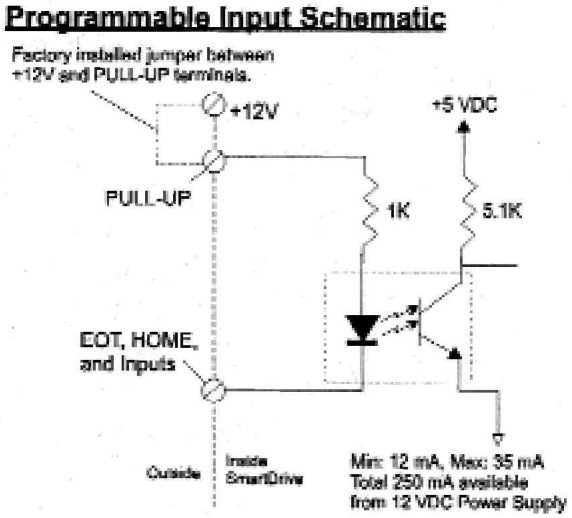
AC Power Input and RPACK Output

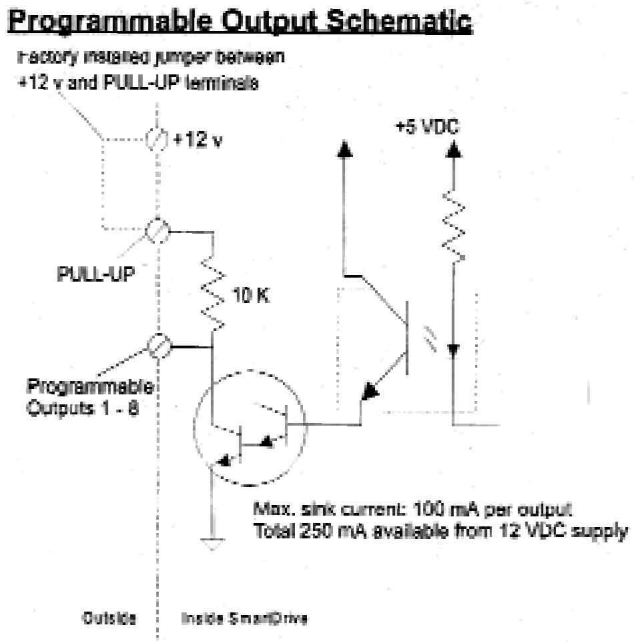


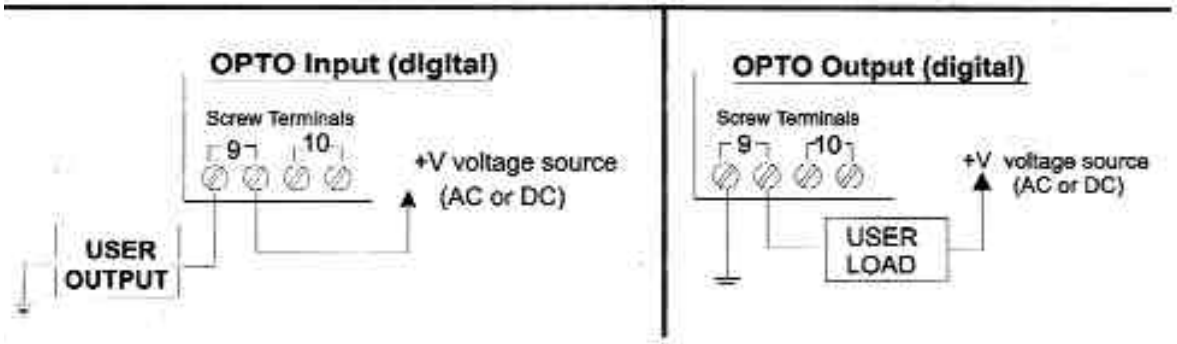
Motor/Encoder Hall inputs



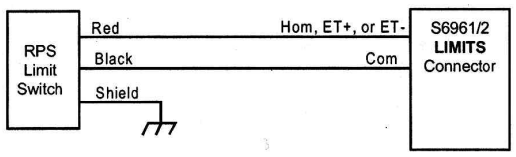
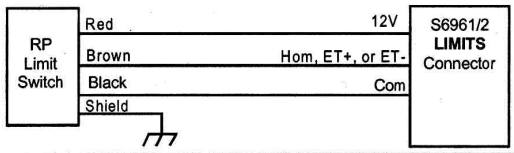
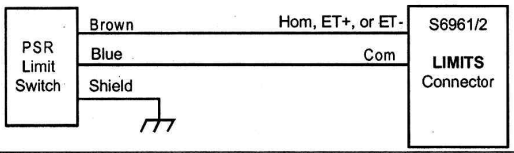
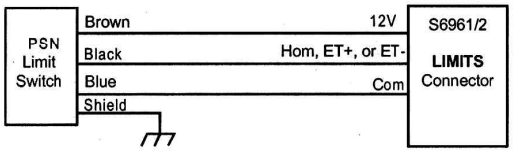
B8961 and B8962 Input and Output Schematics







Connecting IDC Limit Switches to the B8961/2

Switch	Type	Comments	Connections
RPS-1 RPS-2	Normally Open Home only Normally Closed	Reed Switch - has less wiring and lower cost Mounts to N, T, R2, R3, and R4 Actuators	
RP-1 RP-2	Normally Open Home only Normally Closed	Hall Effect Switch - has longer life, and uses LEDs Mounts to N, T, R2, R3, and R4 Actuators	
PSR-1 PSR-1Q PSR-2 PSR-2Q	Normally Open Home Only Normally Closed	Reed Switch - "Q" indicates Quick Disconnect Mounts to EC, NV	
PSN-1 PSN-1Q PSN-2 PSN-2Q	Normally Open Home Only Normally Closed	NPN Hall Effect - "Q" indicates Quick Disconnect Mounts to EC, NV	
PSP-1 PSP-1Q PSP-2 PSP-2Q	Normally Open Normally Closed	PNP Hall Effect Used on Controls requiring PNP Mounts to EC, NV	Will Not Work with IDC Controls

Trouble Shooting

Most problems encountered during the startup and operation of a B8000 Series Smart Drive can be traced to a few basic causes such as mechanical problems, incorrect or noisy signal wiring,attempting to operate beyond the system's capabilities,or software setup or programming errors.A combination of these underlying problems can present an overwhelming variety of symptoms that make it very difficult to track down the root cause of the difficulties.

A logical and methodical approach to trouble shooting is essential to isolating and resolving these problems.The Smart Drives provide a number of tools to make this task easier.The keypad is the most versatile of these troubleshooting tools.The keypad will display error messages,I/O status,motor position,and can be used to run programs in **TRACE** mode to aid in software debugging. Servo Tuner, Application Developer, and terminal programs are also frequently used as trouble shooting tools.

The monitor output port is also a powerful troubleshooting tool.It can be configured to show actual and commanded velocity to help resolve move performance difficulties.Configuring the port to output actual torque can be helpful in determining when a move's torque demands are starting to approach your system's theoretical limits.This mode can also be used to document your machine's torque safety margin.More details are available in the Servo Tuner manual or from Servo Tuner Help.

The following table will help you isolate some of the more common application problems when using B8000 Smart Drive control/drive systems.

Symptoms and Solutions

Symptoms	Probable Causes	Possible Remedies
I/V Fault	Overvoltage due to excessive regen energy from the motor.	Install an RPACK module for additional regen capacity, or reduce the cycle time.
	Line Voltage exceeds 240VAC	Reduce input voltage to allow for surges or spikes caused by other equipment. Replace shorted motor.
	Over current due to a motor short.	Select the correct motor model AND voltage from SETUP/MOTOR/TYPE.
	Over current due to an improperly tuned current loop.	
Thermal Shutdown	Motor winding temperature exceeded 135˚C,or the Smart Drive heatsink temperature exceeded 55˚C	Increase cooling:circulate more air past heatsink,add -FK1 or FK2 fan kit,or reduce duty cycle.
	Missing or damaged motor thermostat	Jumper Temp and Temp Ref together if the motor has no thermostat.
Following Error	Too narrow a following error window.	Set the following error from the SETUP/ENC/FOL-ERR menu.
	Commanded moves too aggressive for system.	Reduce acceleration and velocity.
	Mechanical binding causing excessive following error.	Check for unusual binding.
	Poor or "soft"tuning allowing excessive following error.	Follow the tuning procedure outlined in the Servo Tuner manual,or use the default parameters.

Symptoms	Probable Causes	Possible Remedies
RMS Shutdown	Mechanical jam,amplifier shut down to prevent motor overheating.	Check for binding or reduce acceleration and duty cycle.
	Vertical or constant force application causing false RMS fault.(DSP V1.6 only)	Contact IDC's application dept. Load correct file via Servo Tuner.
	RMS current limit exceeded.	Check for binding or reduce acceleration and duty cycle.
	Wrong motor file loaded.	Select the correct motor model AND voltage from SETUP/MOTOR/TYPE.
"Hit a Limit" error message	Limit switches improperly wired or not connected	EOT inputs must be connected to COM via a jumper wire or a NC switch before motion can take place.
Motor moves the wrong distance	Drive and control resolution don't match	Check the distance units and the Gear Ratio setting in the EDIT/SETUP/MECH menu.
Controller doesn't respond to keypad inputs	The keypad has been disabled	Check the dipswitch settings on the back of the keypad to selectively disable keypad functionality. Check for "keypad disable"input configuration.
Controller doesn't recognize OPTO inputs	OPTO positions default to inputs.	Configure as output from the EDIT/SETUP/I/O menu.
No motion occurs when programs are run	Correct motor type not selected. Motor type defaults to NONE.	Select the correct motor and voltage from the SETUP/MOTOR/TYPE menu.
Homing moves are very slow, don't follow commanded velocity	Home switch active level set incorrectly	Change home switch active level from the SETUP/HOME/SWITCH menu.
Final homing position is not located over the home switch	Homes to the Z-channel on the motor encoder	Move the switch,the motor with respect to the switch,or run in OPEN LOOP MODE.
Motor will not move/has no torque	Drive not enabled.	Check for drive faults with the keypad or the the SS command.
	End of travel limit switches are active	Drive requires Normally Closed limit switches. Inputs should be jumpered if not being used.
	The MOTOR/TYPE selection is wrong.	The default drive configuration is "NONE." Select the correct motor and voltage from the SETUP/MOTOR/TYPE menu.
Motor "jerks"on power up	Hall effect sensors mis-wired	Check feedback cabling
	Wrong motor type selected	Select the correct motor and voltage from the SETUP/MOTOR/TYPE menu.
Motor "whines"	Highly resonant mechanical system	Stiffen the mechanical transmission
	Tuning parameters are set too "hot"	Reduce the gain parameters. K_i is often a strong contributor to this instability.
Motor won't change direction or moves only in one direction	One EOT limit switch is active	Check limit switch wiring or motor/actuator position.
Motor moves in the wrong direction	Control and Drive directions conflict with each other	Change the direction from the SETUP/MOTOR/DIR menu.
Motor runs hot	High accelerations and duty cycles. IDC motors are rated up to a case temp of 100C,and are protected by a thermostat.	Reduce duty cycle.Motor heating is not normally a problem unless it causes Thermal Shutdowns.

IDC Product Support

Factory Authorized Distributors

IDC has factory authorized automation technology distributors located throughout North America,Western Europe,and the Pacific Rim.Each has been selected for their technical expertise,their local market knowledge,and exemplary business practices.They are ready to assist you in applying IDC's systems,as well as other complementary equipment.Contact us at (800) 227.1066 for the name of the distributor in your area.

Regional Offices

IDC Distributors are supported by local,direct Danaher Motion Field Sales Engineers (FSE's).IDC Regional Managers are available to assist with unusually demanding applications,present on site customer seminars,determine custom product needs,or respond to high volume requirements.

Toll Free Technical Support

IDC employs a large staff of mechanical and electrical engineers,whose full time responsibility is to help you select the proper system,install it correctly, and get it up and operating to your satisfaction.The toll free number is (800) 227-1066.Email should be directed to:support@idcmotion.com.

CAD Library

All IDC actuator, motor and gear motor CAD drawings,in a .DXF format,are available to save you valuable design time and prevent transcription errors.These are located on website and on the IDC CD-Rom.

Warranty & Repairs

Danaher Corporation warrants this product to be free of defects in material and workmanship for a period of two (2) years from the date of shipment to the end user. Products that have been improperly used or damaged,in the opinion of Danaher Motion,are not subject to the terms of this warranty.

Danaher Motion maintains a repair facility at its factory in Rohnert Park,California for products manufactured by Danaher Motion.Prior approval by Danaher Motion is required before returning any product for any reason.All returned packages must be accompanied by an RMA# (Return Material Authorization Number).

To obtain return authorization,contact your local IDC distributor or IDC.Please note the following procedure:

1. Obtain the model and serial number of the defective unit.
2. Prepare a purchase order for possible repair cost,in the event that the unit is not warranted.
3. Contact your IDC distributor or Danaher Motion 603-893-0588 for an RMA#.
4. Provide the information describing the nature of the failure.The better the information,the faster we'll have your problem resolved.
5. Ship unit prepaid to:

   Danaher Motion
   600 Martin Ave.,Suite 103
   Rohnert Park,CA 94928

Index

A

AC Power, 133
amplifier
  communication,106
  Disable,8,12
  Enable,8,12
Analog inputs,59
analog opto modules,28
analog output,60
Application Developer,
  Installation,93
  Using,95

B

BCD inputs,59

C

Cable
  Encoder, 4
  Motor, 4
Compensator, 44
Configuring Your System,19
Current rating, 1
custom motor, 21

D

daisy chaining,92
digital opto modules,28
Drive Mounting,131

E

EE system variables,50,52
Encoder, 40
  following error, 23
  resolution,22
End-of-travel, 4
Environmental Specs,133

F

fan kit,101,102
following error, 23
Function Keys, 7

G

Gear Ratio,25

H

Hardware connections,134
Hardware specifications,133
Help,16
Homing
  setup,37
  switch,37

I

If statements,54
Input configuration,28
Input functions,28
Input Schematics,135
Input Test, 5
Inputs,Optos,34

J

Jogging,11
  Setup,34
  Velocity, 37

K

keypad,17
  contrast, 9
  dipswitches, 9
  display, 17
  menu structure,10
  remote mounting, 9
Keypad Hardware Features, 9
keypad operation,5, 7

L

Limit Switch
  Stop rate,40
Load Configuration,100

M

Math,48
memory, 15
Monitor Port,46
Motor configuration,20
Motor Output,133
Motor setup,20
Motor Specifications,106
Motor Type, 3
Motors
  Non-Superior, 3,21
Multi-axis moves,55

N

non-volatile memory, 52
non-volatile system variables,52

O

Opto configuration,32
Opto I/O, 28
Output Configuration,32
Output functions, 32
Output Schematic,135
Output Test, 5
Outputs,Optos,36
Overview, 1

P

Power-up program,39
Programming
  Command Summary, 48
  Conditional expressions,54
  Examples,58
  Overview, 47
  RS-232,91
  variables,49
programs
  copying,8,16
  deleting,17
  editing,8,13
  power-up,39
  trace mode, 12

Q

Quick Start, 3

R

Regen fault,132
RPACK,102
RS-232
  addressing,127
  Command reference,108
  daisy-chain,92
  Immediate commands, 120
  Interface commands, 106
  Programming commands, 108
  running programs,107
  Setup commands, 112
  Tuning,102
RS232,91
RS-232 command listing, 112
RS-232 protocol,91
RS-232 Setup
  RS-232,38
RS-232C Operation,91

S

Scaling,24
  acceleration,27
  distance,24,25
  velocity, 26
Servo Tuner, 45,82
Servo Tuning,43
Setup
  Homing,37
  inputs and outputs, 28
  jogging,36
  menu structure,14
  Motor, 20
  procedure,20
  RS-232,40
Specifications,133
Stop rate,41
System variables,49

T

Test
  move,12
  output,12
  program trace,12
timer, 48,88
Tuning,15,43,44
  gains,44
  Keypad,45
  RS-232,45,81
Tuning Stimulus,46

V

variable count (max.), 49
variables,49

W

warranty, 140
Wiring Diagrams,133

Appendix A: IDC Actuator Ratios

Configuring *inch & mm* Units on Smart Drives Used With IDC Actuators

One of the first steps in setting up an IDC Smart Drive with an IDC linear actuator is to configure the distance, velocity, and acceleration to use meaningful units, probably inches or millimeters. This is done via the RATIO (GR) command. The RATIO is the number of motor revs per distance unit. The Distance Unit used is selected via the SETUP/MECH menu from the keypad or Application Developer. Example: GR5:1 means 5 motor revs per 1 distance unit. Several other examples are available in the Configuring Your System chapter of your Smart Drive manual.

The RATIO is also used to scale the velocity and acceleration numbers when user units/sec or units/sec^2 have been selected from the velocity or acceleration menus.

Until now, since each actuator has its own "revs per inch" ratio, data from the model number had to be interpreted and then a ratio calculated. The following pages reduce that procedure to looking up the inch or mm ratio from a table sorted by actuator model number. Instructions are also included to calculate a ratio for other distance units. Increased positional accuracy is often achieved when using these values, because some ratios aren't exact ("3.5:1" is really 50:14, or 3.571428...).

Methods For Configuring Ratio

There are three methods for entering the ratio information. The keypad is probably the quickest method, if your system includes that option.

1. Smart Drive Keypad [EDIT-SETUP-MECH-RATIO menu]
2. Application Developer [Setup, Axis Menu]
3. Direct RS-232C connection [GR Command]

IDC Actuator Ratios

N, T, R2, R3, R4, NM, RM Series

N Series

	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	***Smart Drive*** **Mechanical Ratio Setting** [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
N-992	1	2	2	2 to 1	20 to 254
N-102	1	2	2	2 to 1	20 to 254
N-152	1.5	2	3	3 to 1	30 to 254
N-202	2	2	4	4 to 1	40 to 254
N-252	2.5	2	5	5 to 1	50 to 254
N-312	3.125	2	6.25	625 to 100	6250 to 25400
N-352	3.571	2	7.14286	100 to 14	1000 to 3556
N-602	6	2	12	12 to 1	120 to 254
N-1202	12	2	24	24 to 1	240 to 254
N-995	1	5	5	5 to 1	50 to 254
N-105	1	5	5	5 to 1	50 to 254
N-155	1.5	5	7.5	75 to 10	750 to 2540
N-205	2	5	10	10 to 1	100 to 254
N-255	2.5	5	12.5	125 to 10	1250 to 2540
N-315	3.125	5	15.625	250 to 16	2500 to 4064
N-355	3.571	5	17.85714	250 to 14	2500 to 3556
N-605	6	5	30	30 to 1	300 to 254
N-1205	12	5	60	60 to 1	600 to 254
N-998	1	8	8	8 to 1	80 to 254
N-108	1	8	8	8 to 1	80 to 254
N-158	1.5	8	12	12 to 1	120 to 254
N-208	2	8	16	16 to 1	160 to 254
N-258	2.5	8	20	20 to 1	200 to 254
N-318	3.125	8	25	25 to 1	250 to 254
N-358	3.571	8	28.57143	400 to 14	4000 to 3556
N-608	6	8	48	48 to 1	480 to 254
N-1208	12	8	96	96 to 1	960 to 254
N-9910	1	10	10	10 to 1	100 to 254
N-1010	1	10	10	10 to 1	100 to 254
N-1510	1.5	10	15	15 to 1	150 to 254
N-2010	2	10	20	20 to 1	200 to 254
N-2510	2.5	10	25	25 to 1	250 to 254
N-3110	3.125	10	31.25	500 to 16	5000 to 4064
N-3510	3.571	10	35.71429	500 to 14	5000 to 3556
N-6010	6	10	60	60 to 1	600 to 254
N-12010	12	10	120	120 to 1	1200 to 254

T Series	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	***Smart Drive* Mechanical Ratio Setting** [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
T-101	1	1	1	1 to 1	10 to 254
T-151	1.5	1	1.5	1.5 to 1	15 to 254
T-201	2	1	2	2 to 1	20 to 254
T-501	5.110	1	5.110442	42432 to 8303	4243 to 21090
T-1001	10.007	1	10.00729	68640 to 6859	6864 to 17422
T-104	1	4	4	4 to 1	40 to 254
T-154	1.5	4	6	6 to 1	60 to 254
T-204	2	4	8	8 to 1	80 to 254
T-504	5.110	4	20.44177	169728 to 8303	16973 to 21090
T-1004	10.007	4	40.02916	274560 to 6859	27456 to 17422
T-106	1	6	6	6 to 1	60 to 254
T-156	1.5	6	9	9 to 1	90 to 254
T-206	2	6	12	12 to 1	120 to 254
T-506	5.110	6	30.66265	254592 to 8303	25459 to 21090
T-1006	10.007	6	60.04374	411840 to 6859	41184 to 17422

R2 Series	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	***Smart Drive* Mechanical Ratio Setting** [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
R2-10T	1	0.33333	0.33333	1 to 3	10 to 762
R2-15T	1.5	0.33333	0.5	1 to 2	10 to 508
R2-20T	2	0.33333	0.66667	2 to 3	20 to 762
R2-31T	3.125	0.33333	1.04167	50 to 48	500 to 12192
R2-35T	3.571	0.33333	1.19048	50 to 42	500 to 10668
R2-120T	12	0.33333	4	4 to 1	40 to 254
R2-102	1	2	2	2 to 1	20 to 254
R2-152	1.5	2	3	3 to 1	30 to 254
R2-202	2	2	4	4 to 1	40 to 254
R2-312	3.125	2	6.25	50 to 8	500 to 2032
R2-352	3.571	2	7.14286	50 to 7	500 to 1778
R2-1202	12	2	24	24 to 1	240 to 254
R2-105	1	5	5	5 to 1	50 to 254
R2-155	1.5	5	7.5	7.5 to 1	75 to 254
R2-205	2	5	10	10 to 1	100 to 254
R2-315	3.125	5	15.625	250 to 16	2500 to 4064
R2-355	3.571	5	17.8571	250 to 14	2500 to 3556
R2-1205	12	5	60	60 to 1	600 to 254
R2-108	1	8	8	8 to 1	80 to 254
R2-158	1.5	8	12	12 to 1	120 to 254
R2-208	2	8	16	16 to 1	160 to 254
R2-318	3.125	8	25	25 to 1	250 to 254
R2-358	3.571	8	28.5714	200 to 7	2000 to 1778
R2-1208	12	8	96	96 to 1	960 to 254

R3 Series	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	*Smart Drive* Mechanical Ratio Setting [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
R3-10T	1	0.1666667	0.16667	1 to 6	10 to 254
R3-15T	1.5	0.1666667	0.25	15 to 60	150 to 15240
R3-20T	2	0.1666667	0.33333	1 to 3	10 to 762
R3-30T	3.000	0.1666667	0.5	3 to 6	30 to 1524
R3-50T	5.037	0.1666667	0.83951	3536 to 4212	354 to 10699
R3-100T	10.000	0.1666667	1.66667	10 to 6	100 to 1524
R3-102	1	2	2	2 to 1	20 to 254
R3-152	1.5	2	3	3 to 1	30 to 254
R3-202	2	2	4	4 to 1	40 to 254
R3-505	5.037	2	10.07401	212160 to 21060	21216 to 53492
R3-1002	10.000	2	20	20 to 1	200 to 254
R3-105	1	5	5	5 to 1	50 to 254
R3-155	1.5	5	7.5	75 to 10	750 to 2540
R3-205	2	5	10	10 to 1	100 to 254
R3-505	5.037	5	25.18519	106080 to 4212	10608 to 10699
R3-1005	10.000	5	50	50 to 1	500 to 254
R3-108	1	8	8	8 to 1	80 to 254
R3-158	1.5	8	12	12 to 1	120 to 254
R3-208	2	8	16	16 to 1	160 to 254
R3-508	5.037	8	40.2963	212160 to 5265	21216 to 13373
R3-1008	10.000	8	80	80 to 1	800 to 254

R4 Series	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	*Smart Drive* Mechanical Ratio Setting [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
R4-10T	1	0.1333333	0.13333	8 to 60	8 to 1524
R4-15T	1.5	0.1333333	0.2	12 to 60	12 to 1524
R4-20T	2	0.1333333	0.26667	16 to 60	16 to 1524
R4-30T	3.000	0.1333333	0.4	24 to 60	24 to 1524
R4-50T	5.110	0.1333333	0.68139	28288 to 41515	2829 to 105448
R4-100T	10.007	0.1333333	1.33430	27456 to 20577	2746 to 52266
R4-101	1	1	1	1 to 1	10 to 254
R4-151	1.5	1	1.5	15 to 10	10 to 254
R4-201	2	1	2	2 to 1	20 to 254
R4-501	5.110	1	5.11044	42432 to 8303	4243 to 21090
R4-1001	10.007	1	10.00729	68640 to 6859	6864 to 17422
R4-104	1	4	4	4 to 1	40 to 254
R4-154	1.5	4	6	6 to 1	60 to 254
R4-204	2	4	8	8 to 1	80 to 254
R4-504	5.110	4	20.44177	169728 to 8303	16973 to 21090
R4-1004	10.007	4	40.02916	274560 to 6859	27456 to 17422
R4-106	1	6	6	6 to 1	60 to 254
R4-156	1.5	6	9	9 to 1	90 to 254
R4-206	2	6	12	12 to 1	120 to 254
R4-506	5.110	6	30.66265	254592 to 8303	25459 to 21090
R4-1006	10.007	6	60.04374	411840 to 6859	41184 to 17422

NM / RM	Motor Reduction	Screw Pitch	Overall Ratio (Mtr Turns/Inch)	***Smart Drive* Mechanical Ratio Setting** [EDIT]-[SETUP]-[MECH]-[RATIO] menu	
				Ratio for **Inches**	Ratio for **mm**
M-992	1	2	2	2 to 1	20 to 254
M-995	1	5	5	5 to 1	50 to 254
M-998	1	8	8	8 to 1	80 to 254
M-9910	1	10	10	10 to 1	100 to 254

Steps for Entering Custom Distance Units (when ratio for inches is known)

Instruction	Example
1) **Select User Units** Select a preferred unit of measure for linear distance. This will be used for programming distance, and can be used for velocity and acceleration as well.	*centimeters*
2) **Determine Overall Mechanical Ratio** Look up actuator mechanical "inch" ratio. Units must be "motor turns/inch."	*6.25* *motor turns/inch*
3) **Convert Ratio to *Turns/User Unit*** Convert "turns/inch" ratio by multiplying or dividing by the same factor you would to convert inches to your preferred user unit.	*6.25 ÷ (2.54 cm/in)* *= 2.4606*
4) **Convert Decimal Ratio to *Ratio of Two Integers*** A. Multiply by the power of ten required to move decimal point to the far right. Note that a maximum of six digits can be entered into from the Smart Drive - it might be necessary to round off the number step 3) above. *This is the numerator of the integer ratio.* B. The power of ten becomes the denominator.	*2.4606×10^4 =* *24606* *10^4 = 10000*
5) **Enter Ratio into Smart Drive** A. Press [EDIT], [SEUP], [MECH], [RATIO] to get to the Mechanical Ratio menu. The numbers from step 4) can now be entered. B. Press [ENTER] after entering the ratio numerator, then [→] to move right and enter the ratio denominator. Press [ENTER] after entering the denominator, then [ESC] to move back one menu.	*24606 to 10000* *(24606 revs = 10000 cm)*
6) **Program the Smart Drive** The Smart Drive is now ready to program in your own User Units. Distances will match teh units configured above.	*DI10.0 GO* *(moves 10.0 cm)*

Appendix B: EMC Installation

Electromagnetic Compatibility (EMC) Installation Guidelines

Introduction

The concept of Electromagnetic Compatibility (EMC),that an electrical device should operate as intended without being disturbed by, or disturbing other electrical devices,has always been an implicit part of every successful industrial control installation.Recent regulation changes,particularly in the European Union, have explicitly and formally defined EMC.This appendix is intended to help professionally qualified OEMs,machine builders,and end users successfully install and operate IDC's single and dual axis servo drives and controls (Models:B8961 and B8962) in compliance with these EMC regulations.

There are two components to Electromagnetic Compatibility.The first,that an electrical device not disturb the operation of other electrical devices,is covered by limits on the power levels of both emitted and conducted electrical magnetic noise.The second component,the ability to operate in the presence of electromagnetic noise,is covered by specifications detailing required immunity levels from electromagnetic phenomena that would normally occur in an industrial environment.

The EMC emissions limits can be divided into two categories.The first, radiated noise,is primarily intended to protect nearby radio communication type equipment.These limits cover a frequency range from 30 MHz to 1 GHz. Poorly grounded motor and power cables are usually the primary cause for high levels of noise in this frequency range.The grounding and cabling recommendations in this appendix give detailed instructions for limiting this radiated noise to safe levels. The second set of emission limits are placed on noise conducted back onto the AC power mains.High frequency switching power electronics,like those commonly found in industrial servo motor drives,can,if not properly filtered,conduct unacceptable limits of noise back onto the AC power lines.This noise could adversely affect other devices.It is worth noting that there is a two order of magnitude safety margin between the allowed noise limits and the required immunity limits.This means that a machine level, rather than component level, application of the following grounding and cabling practices is feasible without compromising the operation of the individual components.

The EMC immunity standards cover a broad range of electromagnetic phenomena including:

- Electrostatic Discharges
- RF electromagnetic fields
- Fast transient and power surges into the AC power supply
- Fast transient bursts on the motor and encoder wiring.

Fundamentally robust electrical design is the best defense against this kind of unpredictable noise,but good grounding practices can help prevent intermittent, difficult to track down,noise related problems.

Since the electromagnetic immunity of the Smart Drive is almost exclusively a function of the design,the installation techniques followed will not positively or negatively impact this built-in immunity.The electromagnetic emissions of the machine,on the other hand,are almost completely a function of the installation

methods used.There are only a few things a machine designer can do to minimize the chance of EMC conflicts between sub-systems in his final product,and between his final product and other products.The first is to use products and components that have a sufficient level of EM immunity to assure that their operation will not be affected by other equipment.In some cases,such as sensitive measuring equipment,the very function of the device requires it to be sensitive to electromagnetic phenomena.A second method is to physically separate the sensitive components from the rest of the machine.In many cases,this is also impossible.A third method is to specify components that have an “acceptable”level of emissions. Properly installed,these components will not affect the operation of other components and machines.In the European Union,what constitutes “acceptable”has been formally defined.

The following pages contain information necessary to install and operate IDC brushless servo Smart Drives in compliance with European EMC regulations (see the attached Declaration of Conformity for specific regulation details).It is certainly NOT necessary to implement all of these procedures to get an IDC Smart Drive up and running.Though proactively addressing EMC concerns is always a good practice,unless the final machine needs to meet some kind of EMC standard,no special precautions are needed in most cases.The Smart Drives and the equipment they normally operate around have sufficient levels of immunity to operate successfully. These installation guidelines were developed as a result of IDC’s own testing process.They are meant only as recommendations.In some applications,EMC regulations may be met without implementing all these suggestions.In others,more extensive shielding and filtering may be needed.As always,it is the responsibility of the person taking the complete machine into service to make sure that it meets all applicable safety and EMC specifications.

The recommendations are divided into 5 categories.The first,Shielding and Separation,covers basic control shielding and panel layout techniques.The second section,Grounding,covers some basic Radio Frequency (RF) grounding rules of thumb.The next section,Physical Layout,is intended to give a specific example of an EMC compliant panel layout.The fourth section covers AC line filtering.In order to meet the European Union conducted emissions specifications a line filter must be installed between the Smart Drive and power mains.The last section details some of the steps needed to install the Smart Drive in compliance with the European Union radiated emissions specifications.Properly shielding and grounding the motor power cable is the most critical step in bringing an EMC compliant system into service.

Separation and Shielding

Since the power of electromagnetic radiation drops off as a square function of the distance from the source of the radiation,proper electrical panel layout an help prevent compatibility problems.Small changes in your panel layouts can make large differences in preventing compatibility problems.Care should be taken in keeping sensitive electrical components,especially sensitive measurement equipment as far away as possible from high voltage,high power electronics such as IDC Smart Drives.Though,before a machine builder goes to extraordinary lengths to separate power and control electronics,it should be noted that IDC servo drives and controls have DSPs and microprocessors operating within an inch of the drive power electronics.

Another source of the “black magic” reputation of electrical noise abatement comes from the area of cable shielding.There are two objectives to cable shielding.The first is to shield the signals in the cable from outside interference.Traditionally, protecting sensitive,low voltage signals,such as encoder and resolver feedback signals,from corruption has been the main focus of industrial cable shielding. Preventing the high voltage,motor power signals from destroying the integrity of these feedback signals is obviously a key part of a functional machine.

The second objective of shielding is to protect the outside environment from unacceptable electromagnetic noise emissions. With the advent of formal electromagnetic noise emission standards,this facet of shielding has become even more important.The high frequency (30 MHz to 1 GHz) spectrum covered by these regulations dictate using at least a quality, braided,shield on all cabling.The small cracks and imperfections found in foil shields make them ineffective at these high frequencies.Even low power cables must be properly protected with a braided shield.These low voltage,signal or feedback wires often share a logic ground with the microprocessor and clock of the control.The clock ground is a significant contributor to the radiated noise in the frequency range in question.

Another way in which EMC shielding differs from traditional shielding is in the connection of the shield to ground.A solid,360° connection to ground rather than a single wire connection or even a braided strand connection can help prevent radiated noise problems.Connecting the cable shield to ground at BOTH ends of the shield can also help reduce the level of radiated noise.Please note there could be safety problems with this practice if the two grounds are at different potential levels. The shield could actually become a current carrying conductor. Safely connecting shields should always take precedence over any potential radiated emission reductions.

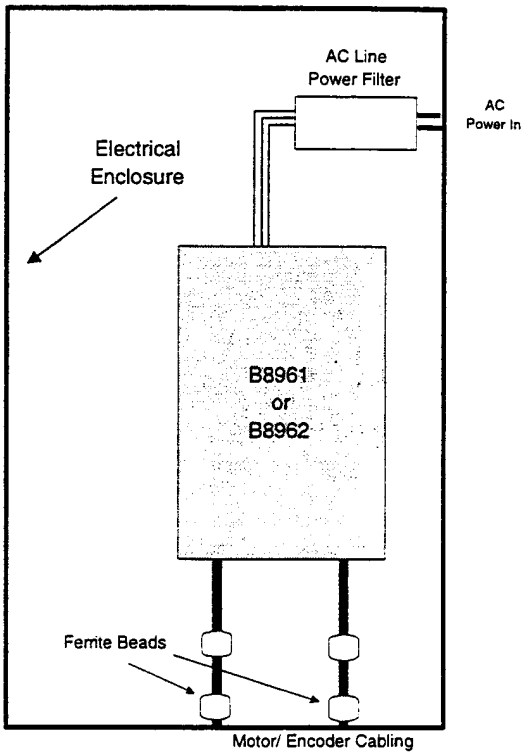
Grounding

Grounding is another topic that has a slightly different meaning in an EMC context than it has had traditionally. Generally, local wiring safety codes thoroughly cover the subject of grounding,as it relates to personal safety, but they rarely address proper high frequency grounding practices.A good safety or DC ground can be made with a wire or small area contact.A high frequency ground connection requires a broad contact region or a wide braided strap to be effective.Simple wires can act like an open circuit at the radio frequencies under consideration. For this reason when mounting component or chassis to a cabinet,it is important that the paint on both surfaces be scraped away.A simple star washer grounding scheme may not be effective.

As always,the relevant local safety standards supersede any EMC precautions.

Physical Installation

The drawing at right shows an example of an EMC compliant IDC servo Smart Drive installation.It is assumed that the AC line filter and the Smart Drive have a good RF ground connection to the cabinet frame,and that the cables are shielded and grounded at both ends. The guidelines outlined above for grounding and cable separation should be followed when extending the concepts illustrated here to multiple component enclosure mounting.



As always,it is the responsibility of the machine builder to properly install the IDC servo drives and other electrical components making up the machine before taking the machine into service.When multiple electrical components are installed together on a machine,the radiated and conducted noise emission levels can be both greater or less than the levels of the single component.The immunity levels of the individual components will not be affected.

AC Power Line Filtering

As mentioned above,one of the characteristics of the high power electronics used in servo drive amplifiers is the broadband noise their switching conducts back on to the AC input line.There are a couple of ways to reduce the magnitude of the noise on the AC line.The first is to reduce the switching speed of the drive electronics,but this would increase the amount of heat the power transistors would need to dissipate.This in turn would increase the size and cost of the servo drive.

A second way would be to build an AC filter and/or isolation transformer into every drive.This solution also increases the size and cost of every drive.A third solution is to install an AC line filter at the point where the AC power enters the protective enclosure where the servo drives and other machine control electronics are mounted.This is the method documented and used by IDC controls.The main advantage is that the cost and size of the filter can be specifically tailored to the application at hand, rather than being sized for a “worst case scenario”like an internal filter circuit.A second advantage is that this externally mounted filter can be sized to supply multiple drives or other machine control electronics.

The proper installation and sizing of the AC line filter is critical in attaining acceptable attenuation of the broad band noise caused by the switching power electronics.

IDC has tested and documented the effectiveness of the Corcom EMI filters (part #: 6EQ8 F7264) but any filter with a similar power and high frequency impedance characteristics should provide similar results.

The proper installation of the line filter is at least as important as properly sizing it to the application.The following guidelines should be used:

1. Ground the line filter case to the enclosure cabinet ground.The case of the line filter is generally conductive and connected to the filter's ground.Insuring a good RF connection between the filter and the cabinet usually requires removing the paint where the filter is mounted to create a large surface area conduction path.
2. Minimize the AC cable run inside of the cabinet.The filtered side of the AC line filter can be contaminated by radiated noise from the unfiltered side of the filter and from other RF sources inside of the cabinet,thus completely negating the effect of the filtering.The best way to do this is to mount the filter at the point where the AC power enters the enclosure.

Motor Output Power Cable Shielding

Most of the radiated RF emissions come from the motor power cables.Use shielded motor cables.The shield needs to be connected at the motor AND the cabinet ground as close to the drive as possible for best results.Removing the paint around the bottom drive mounting tab,and using this bolt as the shield termination generally works very well.To fully comply with emission standards set forth in EN50081-1 (Residential,commercial and light industrial environments) the motor power cables also need to be run inside of conductive conduit.The conduit should make good electrical contact with the motor case ground at the motor end,and the enclosure ground at the motor end.Please note that local electrical codes may have further rules and restrictions on high voltage and current cabling.These installation instructions are intended to supplement,not replace or supersede the local electrical, fire,and safety codes.

In addition to grounding both ends of the motor and encoder shield,to further reduce the radiated emissions from the drive, ferrite beads can be installed on the motor and encoder cables before they leave the electrical enclosure (see the drawing above).IDC has tested and confirmed the effectiveness of the Fair-Rite model #0443164151.This is a removable "snap-on"style emission suppressor.These beads are necessary to meet the "residential and light industry"emission specifications.Other specification levels may or may not need the additional emission suppression provided by the beads.And in some cases,other measures may need to be taken.

DECLARATION OF CONFORMITY

For declaration of conformity, please contact the factory (1-800-227-1066 or 1-603-893-0588).

Appendix C: Fan Kit Option

IDC controls are designed for convection cooling.The shape and size of our heatsinks are the result of thermal analysis and experimentation.

All of our controls have built-in temperature protection.Thermal sensors inside the B8000 controls will activate at a conservative heatsink temperature of 55°C.Thus, IDC controls will not be damaged when overtemperature conditions occur.

A number of factors affect the internal temperature of a control and whether or not it needs additional cooling:
- Ambient temperature
- Air flow
- Duty cycle
- Power delivered (the RMS current output)
- Number of axis per control
- Regenerative energy returned from the load
- Bus voltage (B Series)

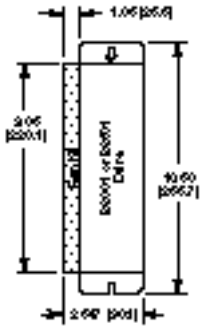
Adequate ventilation in the enclosure does a lot to cool our controls.Most often, a single fan in your enclosure or panel will circulate enough air.

When is a Fan Kit Needed?

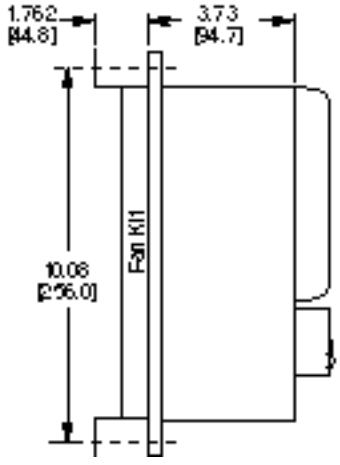
Here are a few general guidelines to indicate when a fan kit may be necessary for your high performance application.

B Series Controls:
- With high regenerative loads that do not require an RPACK.
- Vertical,high friction,or clamping applications.

Dimensions in [mm]
Front View (B8001, B8501)



Side View (B8961,B8961)



Model	A
B8961	3.73 [94/7]
B8961	6/79 [179.51]

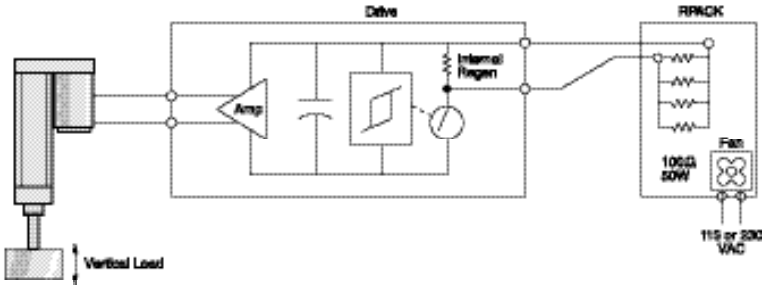


Appendix D: Regen Pack Option

When a large inertial load is decelerated or a vertical load is lowered, the mechanical energy that is not dissipated as heat in the actuator or drive is "regenerated" by the motor and transferred back into the drive's power supply. This causes the drive's power supply voltage to increase. Without circuit protection, this voltage increase can damage a drive. Without a means of dissapating this energy, such applications cannot be solved.

All IDC servo drives are fully protected against excessive regenerative energy. First, they are overvoltage and short circuit protected. Second, they are capable of dissipating regenerative energy both internally, and in extreme cases, externally using our model RPACK-1 or RPACK-2. Our drive's LEDs will even indicated when excessive regenerative energy is present in your application.

An RPACK allows you to make more aggressive moves in high inertia, low friction applications. Each RPACK provides connections for hook-up to either servo or stepper drives for a total dissipation of 240W continuous and 1000W peak (for 3 seconds).



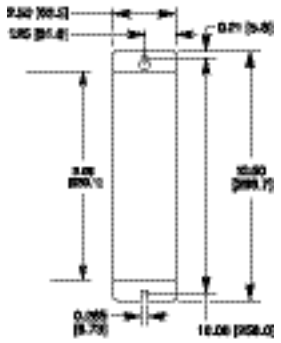
Regen Capacity (watts or joules)	B8000 Series
Internal without fan	Total energy storage capacity (max)
Continuous	70 joules (1 axis)
Peak (3 sec)	140 joules (2 axis)
Internal with fan	
Continuous	70 joules (1 axis)
Peak (3 sec)	140 joules (2 axis)
RPACK (additional)	
Continuous	240 watts
Peak (3 sec)	1,000 watts
When to use an RPACK-1 or RPACK-2	If drive faults due to overvoltage If load is vertical with ballscrew If decelerating large inertial loads

Dimensions in [mm]

Front View



Side View



Danaher Motion, Precision Systems Group
7C Raymond Avenue, Salem, NH 03079
(800) 227-1066 • FAX (603) 893-8280
OUTSIDE THE U.S. CALL (603) 893-0588

E-mail: sales@idcmotion.com
World Wide Web: http://www.idcmotion.com
B8961/B8962 Operator's Manual, PCW-4744 Rev.1.9
July 2003

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP