Industrial
Devices
Corporation

Indexer

961 Single Axis
962 Dual Axis

User's Manual

Rev. 1.1 **PCW-4908**

Revision History

Version 1.0 - July 17, 1997
Revision 1.1 - May 1, 1998

Industrial Devices Corporation (IDC) strives to maintain effective communication with all users and potential users of our products. If you have any questions or concerns regarding this product and technical manual, or any of our products, please contact:

Industrial Devices Corporation
64 Digital Drive
Novato, CA 94949

TEL: **(800) 747-0064** FAX: **(415) 883-2094**

OUTSIDE THE U.S. CALL **(415) 382-4300**

WEBSITE: **www.idcmotion.com**

EMAIL: info@idcmotion.com

-

Table of Contents

Overview of Product and Manual

IDC's 961 and 962 Indexers integrate a motion/machine controller, digital and analog I/O, and an AC power supply in a single compact unit. The optional FP220 keypad operator panel can be mounted directly on the Indexer or connected via the remote cable (included). The 961 is the single-axis version, and 962 is the dual-axis version.

IDC's 961/962 Indexers provide excellent value when your application calls for any of the following:

- Mixed digital-servo and stepper-motor control
- A basic 1or 2 axis integrated machine controller
- A cost effective operator interface (keypad)
- On-board OPTO I/O modules (analog or digital)
- Flexible I/O capabilities
- Applications using non-IDC drives or motors
- Thumbwheel data inputs

The Indexers provide unmatched flexibility when teamed-up with the optional keypad operator interface. We recommend the keypad for users of all experience levels because it facilitates the most efficient possible configuration and programming of applications. Initial configuration and programming, in most cases, will require only a few minutes using the keypad. In addition to being fully programmable from the keypad, the 961/962 can also be programmed over RS-232C with IDC's user-friendly Application Developer software. RS-232C commands and setup definitions are in chapter 5.

This manual has been designed to help you successfully install, program, and operate your Indexer. Each section of the manual emphasizes the common theme "ease of use". If you have any questions that are not adequately answered in this manual, please contact our factory application engineers at (800) 747-0064.

The *Quick Start* section will help you quickly confirm basic system operation. Chapter 1, *Using the Keypad*, describes communicating with the Indexer.

Chapter 2 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 Chapter 2.

Chapter 3, *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 easy to remember and intuitive, without sacrificing flexibility or power.

In Chapter 4, *IDeal™ Command Reference*, an alphabetical list of Smart Drive commands explains syntax, ranges, defaults, and provides programming examples in detail.

Chapter 5, *RS-232C Operation*, is for users who plan to configure and program the Indexer in an RS-232C-hosted mode. IDC's *Application Developer* program follows a standard Windows dialog-box structure to make configuring and programming the Indexer very straightforward. This section also covers RS-232C command syntax and definitions for users who are not using Windows.

The *Hardware Reference* chapter provides detailed I/O schematics, cables, and specifications. Included with this manual is the *IDCMotion*™ disk set. IDC's Application Developer and Servo Tuner software are automatically installed on your hard drive by running the setup program on Disk 1 of the *IDCMotion*™ disk set. This disk also includes a readme file containing the latest information on software features. The readme file also contains a program listing of demo programs included with *Application Developer.*

Shipping Contents

Your 961/962 Indexer will arrive equipped as listed below. If any parts or accessories are missing, please call IDC Customer Support at: (800) 747-0064.

961 or 962

If you ordered part number 961 or 962, you will receive, in addition to the Indexer:

1. Keypad (FP220)
2. Remote cable for Keypad
3. Keypad mounting gasket
4. Keypad mounting template
5. IDC*Motion* software
6. AC power cord
7. IDC screwdriver
8. Indexer mounting bracket, installed on the back of the unit for minimum depth mounting

961NP or 962NP

If you ordered part number 961NP or 962NP, you will receive, in addition to the Indexer:

1. AC power cord
2. IDC*Motion* software
3. Indexer mounting bracket, installed on the side of the unit for minimum width mounting

Note: Because the 961NP and 962NP arrive ready for minimum width mounting, customers often order a combination of the NP Indexer plus the FP220 Keypad option.

FP220 Keypad and Accessories

If you ordered the FP220 Keypad, you will receive the following:

1. Keypad Operator Panel
2. Remote keypad cable
3. Keypad mounting gasket
4. Keypad mounting template
5. 3 ball-head mounting screws for mounting keypad on the Indexer
6. IDC screwdriver

Important Information for the Indexer User:

- ❒ Your Indexer has been shipped from the factory with the **Step/CCW** jumpers in the **Step** position, and with the **Differential/Single-Ended** jumpers in the **Differential** position.
- ❒ **If you need to change these jumpers,** please see pages 111 and 112 in Chapter 6 - Hardware Reference.
- ❒ Remember, **all jumpers must be changed** when you perform this modification.

Quick Start

The purpose of the Quick Start is to help an experienced motion control user verify that the 961 or 962 Indexer is operable and ready for configuration and programming. The following directions assume that the user is familiar with motion controls, and their related electrical connections.

The IDC Keypad (FP220) is highly recommended for the easiest possible setup and programming of your application. In this section it is assumed that a keypad will be used, even though an RS-232C connection may be used later.

Apply Power

1. Connect keypad to Indexer by remote cable as shown below, or mount keypad directly onto the Indexer on the mounting posts provided.
2. Connect power cord to the 120 VAC connection on the end of the Indexer (shown below). Power is applied to the Indexer when power cord is plugged into the AC power source. There is no ON/OFF switch on either the keypad or Indexer.

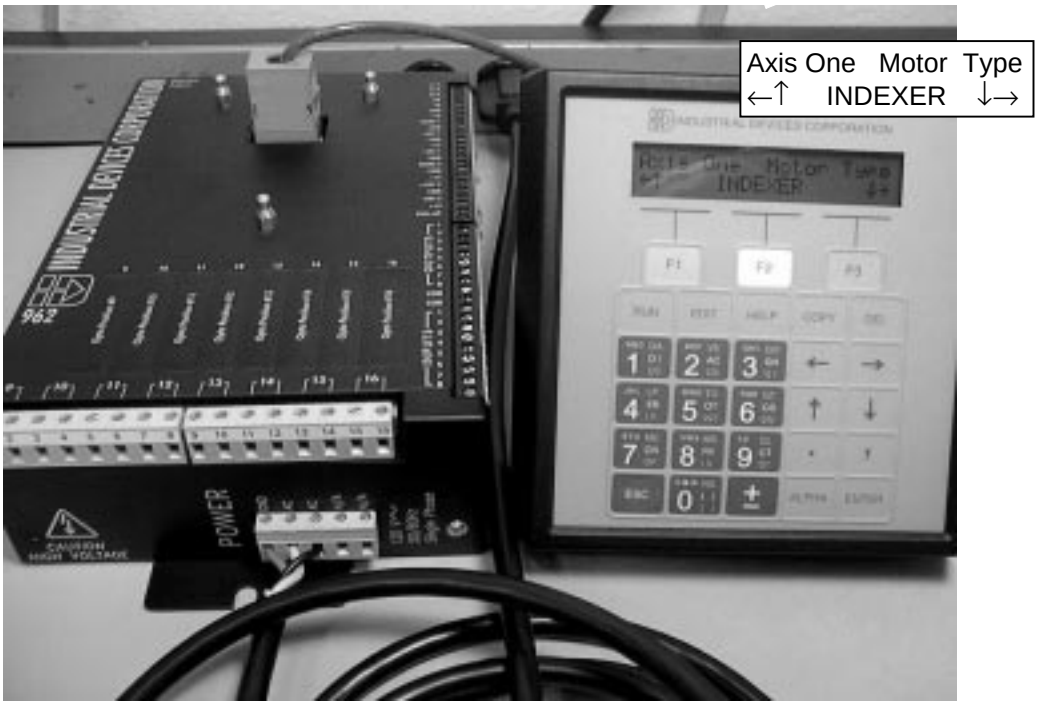
Verify Operation

When power is applied, the keypad LCD display briefly shows Model # and Firmware Revision, then changes to the Main or Diagnostic Display as shown to the right. The Main Display 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.

Axis 1	Axis 2
+0.0000	+0.0000
00000000	00000000
Inputs	Outputs

Note: The LCD display may need adjustment for better viewing. If so, please refer to "Contrast Adjustment" in Chapter 1- Using the Keypad.

☞ On the keypad, press **EDIT** > **SETUP** > **MOTOR** > **TYPE**. The main display should read **INDEXER** as shown below. This verifies that your Indexer is operable and you may begin configuring and programming the system for your application.



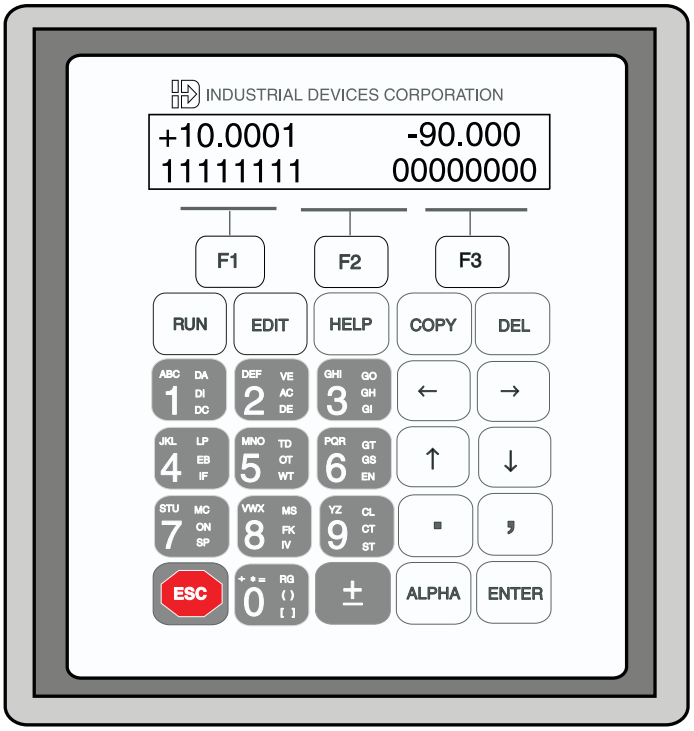
Chapter 1 - Using the Keypad

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 general categories. It functions first as a programming and troubleshooting tool, and secondly 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 Structure*, gives the programmer a broad overview of how the keypad's setup and programming menus operate. Detailed information about each setup parameter is presented in *Chapter 2 - Configuring Your System.*

Keypad Features

- Easy-to-read, 2-line, 40-character, back-lit supertwist display
- Can be sealed to IP65 (NEMA 4) washdown environment
- Large, scratch-proof keys with audible and tactile feedback
- Attaches to Indexer with remote cable or mounts directly on Indexer



Descriptions of Individual Keys

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, lists programs, and resets position counter.

HELP

Provides help on keys, menus and command syntax.

COPY

Copies one program to another within a unit.

DEL

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

Arrows (← , →, and ↑ , ↓)

For scrolling 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, F2, and F3 to enter commands in the editor.

Keypad Hardware Features

Setting Dipswitches

Four dipswitches on the back of the keypad provides a means of preventing access to certain keypad menus. If access to a menu is denied, pressing that menu key will have no effect. For example, if **1** is **ON**, and **2** is **OFF**, the operator will be able to stop motion by pressing the **ESC** (escape) key, but will not be able to access the **RUN** menus to select another program. This is a hardware inhibit, and is independent of any software or setup parameter in the Smart Drive. See the chart below for switch assignments.

Dipswitch Settings				Function Level
1	2	3	4	
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

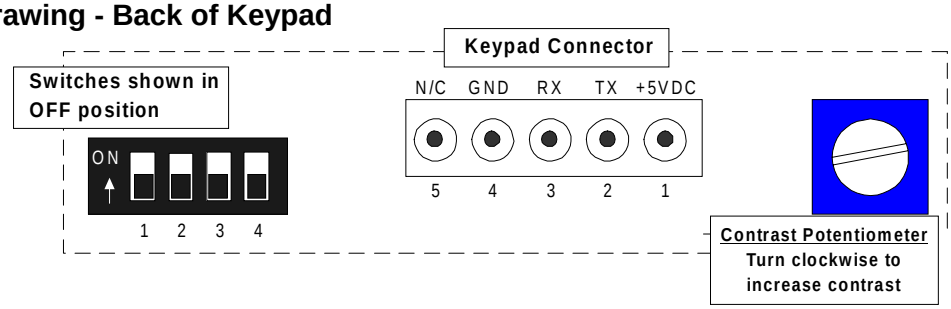
* Reserved for future functions.

Notes: Power must be cycled before dipswitch changes take effect. Access to the JOG menu can be enabled or disabled from software.

Adjusting Contrast

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

Drawing - Back of Keypad



Remote-Mounting the Keypad

The keypad can easily be 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 toward the mounting enclosure (not the keypad). A pressure seal is formed between the gasket and the keypad, while the adhesive maintains a seal between the enclosure and the gasket. The keypad communicates with the Indexer via RS-232C and this cable may be extended if necessary. At longer distances, users may be required to provide a separate 5 VDC supply to power the keypad.

Note: Complete keypad drawings and pin-outs can be found in the *Hardware Reference* chapter. An FP220 Keypad Mounting Template is included with every keypad and may also be found in the *Hardware Reference* chapter.

Run-Time Operator Interface

While keypad programming and system configuration are defined by IDC, run-time operation (how the machine operator interfaces with the Indexer) falls completely within the customer's control.

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

- 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 Indexer's operator interface, see the *Programming Your Application* and *IDeal Command Reference* chapters.

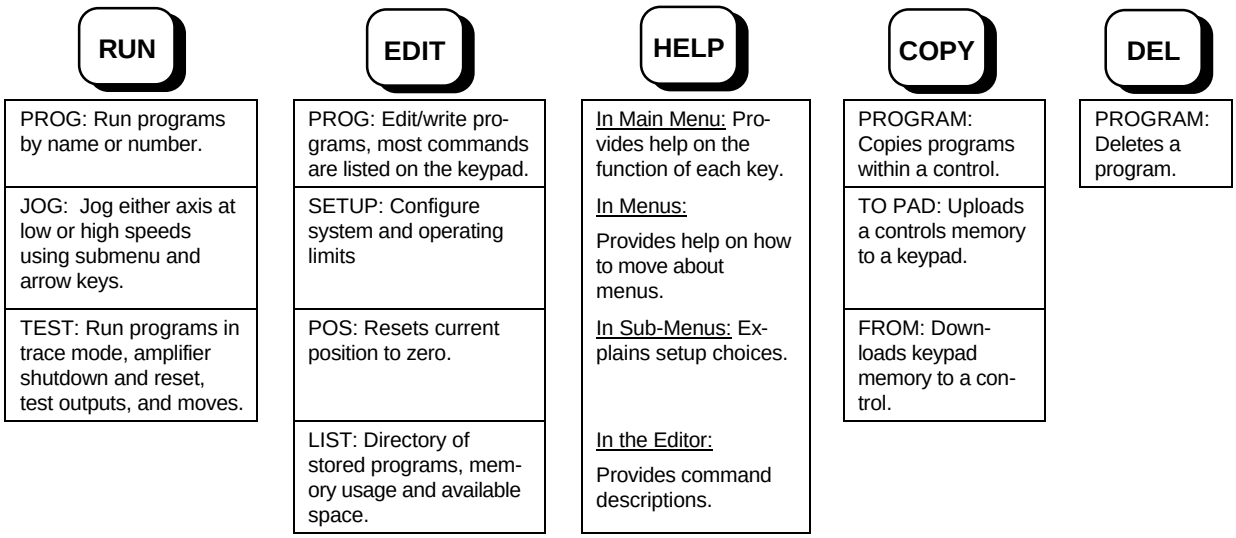
Keypad Menu Structure

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 displayed above one of the function keys, F1, F2, or F3. Press a function key to 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. Press the appropriate arrow key to cycle, one display at a time, through all options in that menu. To leave a menu, without making a selection, or to back up one menu level, press ESC.

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

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



Using the RUN Menus

RUN

Pressing the RUN key displays a set of sub-menus. Access the sub-menus by pressing F1 (PROG), F2 (JOG), or F3 (TEST). Following are instructions for various activities within each sub-menu:

Keypad Display

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

F1 F2 F3

Select PROG To Run a Program

To run an existing program by the program number:

```
↑ RUN PROGRAM ↓
> 5
```

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

To run an existing program by name:

```
--↑ RUN PROGRAM ↓-
>12 GRIND
```

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

Select JOG To Jog the Motor

To jog either axes of the Indexer:

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

1. Press RUN
2. Press JOG (F2)
3. Press ← and → keys to move axis 1
   ↑ and ↓ keys to move axis 2 on a 962.

Note: The ↑ and ↓ keys will also jog Axis 1 on a 961.

Change between LO and HIGH speeds with the F1 and F2 keys. Jog speeds and accelerations can be changed in the EDIT > SETUP > JOG menu.

To jog an incremental distance:

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

1. Press RUN
2. Press JOG (F2)
3. Enter the desired distance number (i.e., 0.012).
4. Press and release an arrow key to make the motor move this distance. The arrow pressed determines the direction of the move.
5. Repeat 3 and 4 until desired position is reached.
   Repeatedly pressing the arrow keys will jog the same distance until a new distance is defined. This feature is intended for very fine, final positioning. The incremental jog speed is therefore fixed at a *very* low speed.

Note: Pressing ESC at any time will terminate the incremental JOG mode.

RUN Sub-Menus continued:

Select TEST For Testing and Debugging

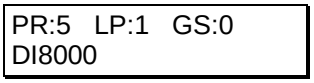
TEST Sub-Menus

TRACE

The trace feature allows you to debug programs by sequentially executing one program command at a time.

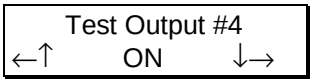
1. Press RUN > TEST > TRACE
2. Enter the program name or number
3. Press ENTER

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 Indexer's outputs, as well as the devices to which it is connected, by forcing them on and off.



1. Press RUN > TEST > OUTPUT
2. Press ← → keys to scroll through outputs 1-8, and any OPTO positions configured as outputs.
3. Press ↑ ↓ 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

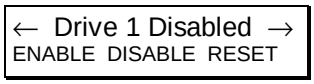
This selection moves your motor shaft one user-defined unit forward and backward. This allows you to verify basic motor, encoder, and amplifier operation.

1. Press RUN > TEST > MOVE
2. Press F1, F2, or F3 to select the axis to move.

RUN Sub-Menus continued:

SHUTDN (Shutdown)

Selecting SHUTDN allows you to enable, disable or reset axis 1 or axis 2 (962). When a drive is disabled, the amplifier is off and your motor has no power. The shaft can easily be manually rotated. RESET returns each drive to its power-on condition. Existing set-up parameters are restored.



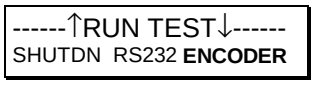
1. Press RUN > TEST.
2. Press ↑ or ↓ until SHUTDN appears above F1.
3. Press SHUTDN.
4. Press desired action, ENABLE, DISABLE or RESET.

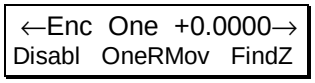
RS232

To be implemented in a future version of software

ENCODER

The ENCODER sub-menu allows you to perform three different tests to determine if encoders are working properly.





1. **Disabl/Enable** allows you to disable the amplifier and manually turn the motor shaft. As you turn the shaft, you should see position changes on the first line of the display, i.e. plus (+) positions in one direction and minus (-) positions the other direction.
2. **OneRMov** allows you to command the motor to turn one full revolution in either direction. Selecting **OneRMov** gives you two choices: **EXTEND** moves the shaft in one direction, and **RETRACT** moves it back.
3. **FindZ** commands the motor to rotate until it finds the Z pulse position. This allows you to accurately center the position of the Home switch.

<u>To access encoder tests</u>:

1. Press RUN > TEST.
2. Press ↑ or ↓ until ENCODER appears above F3.
3. Press ENCODER.
4. Press desired encoder test (described above).
5. Use the ← or → arrows to select the axis of the encoder to be tested.

Using the EDIT Menus

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

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

Pressing the ↓ or ↑ 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 menus by pressing the appropriate function key. See descriptions of each of these sub-menus below.

Using the PROG Sub-menu to Create and Edit Your Motion Control Programs

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

To Create a New Program:

1. Press EDIT
2. Press F1 (PROG) and you will see a display with a blinking cursor as shown to the right.

```
-- ↑   EDIT PROGRAM   ↓ --
>_
```

3. Enter an identifying number that will be used later to call-up the program. The number you enter may be from *1-199, but you must select a number that has not already been used for an existing program. If the Indexer contains several programs, you may need to scroll the list (using ↑ or ↓ keys) to determine a number that has not been used.
   Note: You may assign a name, rather than a number, to your program if you wish. See "Naming Your Programs" earlier in this chapter.
4. Press ENTER. You will see a completely blank display screen with only a blinking cursor in the upper left corner. The Indexer is now ready to accept a program.

```
_
```

5. Once inside the program editor, you will enter commands by pressing a function key and then a numeric key. Examples of creating, saving, naming, and editing programs follow:

Entering Commands with the Keypad

Examples using the number 2 key:

- To enter VE (the upper command), press F1 (blue), then press the 2 key.
- To enter AC (the middle command), press F2 (yellow), then press the 2 key.
- To enter DE (the lower command), press F3 (green), then press the 2 key.
- Press ENTER to insert a space before entering the next command.

Step-by-Step Example of Entering a New Program

To enter the program **AC.3 VE2 DI1 GO**:

1. Press F2.
2. Press the decimal or period key (adjacent to the #9 key).
3. Press the #3 key.
4. Press ENTER.
5. Press F1.
6. Press the #2 key.
7. Press ENTER.
8. Press F2.
9. Press the #1 key.
10. Press ENTER.
11. Press F1.
12. Press the #3 key.

You will see the display shown to the right.

```
AC.3 VE2 DI1 GO_
```

To Save the Program;

1. Press ESC (red octagon in lower left corner or keypad). You will see a "Save Program" query as shown to the right.
2. Press F1 (YES) or F3 (NO).

```
Save Program 23?
 YES                NO
```

To Edit an Existing Program:

Follow the same steps as in "To Create a New Program" above, and remember that:

- Inside the editor, pressing ENTER inserts spaces, which are used as delimiters for commands
- Pressing DEL deletes characters
- The left and right arrows (← or →) scroll through programs one character at a time
- The up and down arrows (↑ or ↓) scroll through programs one line at a time.

Naming Programs

Any or all of the programs stored in the non-volatile memory of the Indexer can be given descriptive names in addition to the program number that the Indexer assigns it. Program names must be put inside of square brackets, [program name], at the start of a program. The name can be up to 14 characters, but the first 10 must be unique. They can, like variables, be nearly any combination of characters.

Programs or subroutines are often named to help "self document" a program. It is usually easier to remember and understand a name then a number. You may call a program or branch to them by name.

This feature also makes it easier for operators to run programs and easier for the programmer to develop systems requiring operator interfacing with our keypad.

Suppose your system will run 20 different parts and each part has a different program. With an Indexer, all you have to do is name each of your programs so an operator will easily recognize them. When the keypad RUN key is pressed, instead of entering a number, simply scroll through the list of program names (possibly part names) using the ↑ and ↓ keys. When the desired part is displayed, simply press ENTER to run the program for that part.

Entering Other Characters with The Alpha Key

The ALPHA key allows you to enter almost any character into a program from the keypad. You will find this desirable if you want to:

- Name your programs or subroutines
- Call your subroutines by name
- Make variable names descriptive
- Use operator messages or prompts
- Send messages over RS-232C
- Use one of the few commands not on the keypad, such as EA or " "

You can insert any of the letters or characters above the numbers on the numeric keys. For instance, the A, B, or C above the #1 key. To insert a character, press ALPHA, then press the numeric key with the character you want. Press the numeric key more than once to select second and third characters. Pressing the numeric key 4, 5 , and 6 times will access the lower case letters. Enter the next character by pressing ALPHA again. Press the ← or → arrow keys to move the cursor to the next space. Press ENTER to move the cursor more than one space. For example, if you want to leave more than one space between words in a message to an operator.

Example of naming a program

Name the program created earlier [MINE].

To insert [MINE],

1. Press F3.
2. Press 0 (zero) key.
3. Press ALPHA.
4. Press 5 key.
5. Press ALPHA.
6. Press 3 key three times.
7. Press ALPHA.
8. Press 5 key two times.
9. Press ALPHA.
10. Press 2 key two times.
11. Press the → key to move cursor to the right of the bracket.

```
[MINE] AC.3 VE2 DI1 GO
```

Using the **EDIT** sub-menus continued:

Using the SETUP Sub-Menus for Configuring Your System

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

Sub-Menus	Setup 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
	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
RS-232C	ECHO	Echo characters
	UNIT#	Serial address
MISC	DISPLAY	Display mode
	STOP-RATE	Decel rate when stop input activated
	TEST	Enable Test Menu
	FAULT	Polarity - Active High or Active Low
	ENABLE	Polarity - Active High or Active Low

using the Edit **Sub-Menus continued:**

Select POS to Reset the Current Position to Zero

POS is a quick way to reset the motor's current position to (absolute) zero - a very useful setup and debugging tool.

1. Press EDIT > POS (F3). You will be queried as shown.
2. Press YES (F1) or NO (F3)

```
Reset Position?
YES                    NO
```

Select LIST to View Program Memory Usage

LIST provides a way to view your program memory usage. Standard program storage in your IDC Indexer is 6K bytes, and the maximum size of a single program is 1,024 bytes. If you chose the 30K expanded memory option, your Indexer will store up to 400 programs, with a maximum single program size of 1,024 bytes.

1. Press EDIT > ↓ > LIST to display the number of programs stored in your Smart Drive.

```
DIRECTORY ↑MORE↓
PROGRAMS: 18
```

2. Press ↓ to display the total amount of memory your programs have used.

```
DIRECTORY ↑MORE↓
BYTES USED:  1186
```

3. Press ↓ to display the number of bytes of memory you still have available.

```
DIRECTORY ↑MORE↓
BYTES FREE: 4958
```

4. Pressing ↓ continuously will take you through the list of programs, displaying the number of bytes being used by each program.

```
DIRECTORY ↑MORE↓
5  <untitled>: 56 bytes
```

TUNING Available only on Brushless-Servo Smart Drives

TEACH To be implemented in a future version of software

Using HELP

If you have a question while 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, press ESC to return to the menu.

Pressing HELP in the Main Menu

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

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

HELP **Menus continued:**

Pressing HELP in Menus and Sub-Menus

HELP explains the selections available from your current menu location.

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

Pressing HELP In the Program Edit function

HELP provides a brief alphabetical list of commands. 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.

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

Using Copy

Copying programs from one name (or number) to another can save you a significant amount of time when programming your Indexer with a keypad.

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

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

Select PROG to Copy One Program to Another Program

This selection allows you to copy any existing program to a new program name.
<u>To copy one program to another</u>:

```
↑ 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).

1. Enter the target program number.
2. Press ENTER

```
↑ TARGET PROGRAM ↓
> 5
```

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

TO PAD To be implemented in a future version of software**.**

FROM To be implemented in a future version of software**.**

Using DEL to Delete Programs or Single Entries

The DEL key allows you to delete any motion program currently in your Indexer.

To delete a program:

1. Press 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. Press ENTER

```
↑ DELETE PROGRAM ↓
> _
```

A single entry (letter or number) may be deleted as follows:

1. Move the cursor over the entry you wish to delete (move with ← or →).
2. Press DEL.

Chapter 2 - Configuring Your System

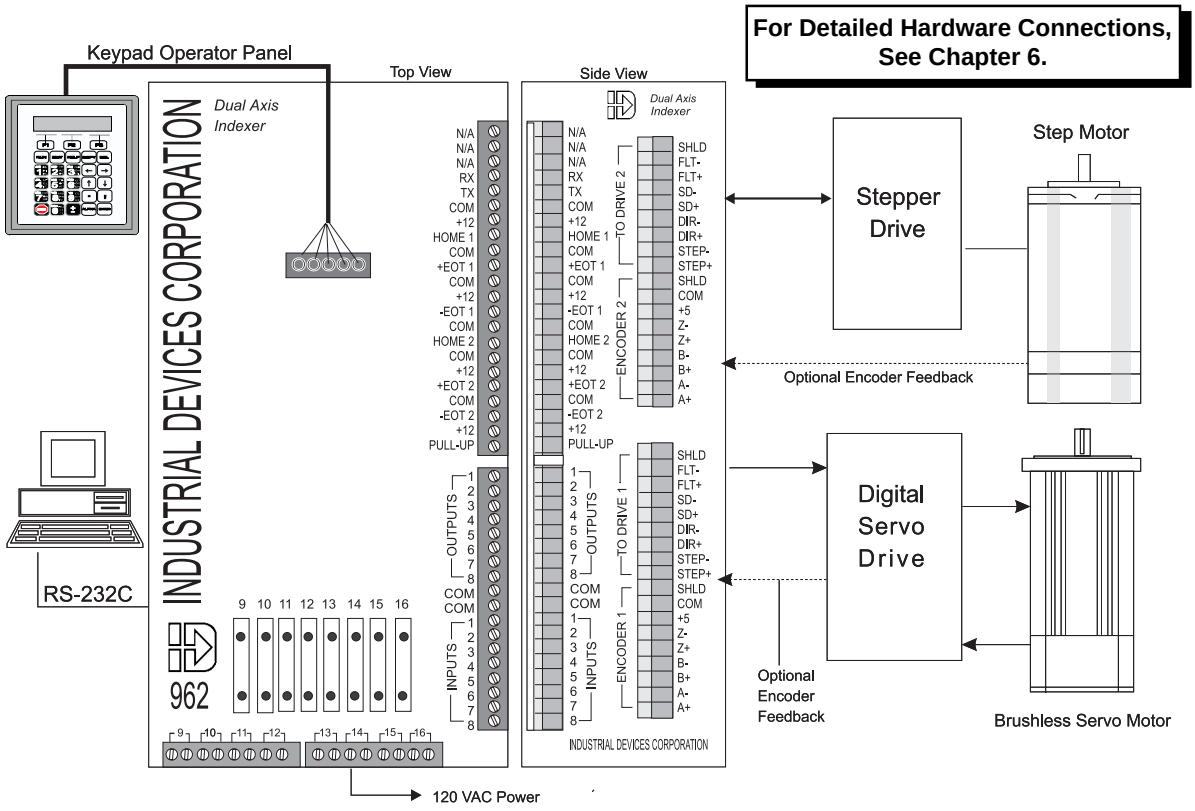
This chapter presents a straightforward procedure for configuring your software to your specific equipment and application requirements. The task of configuring your 961 or 962 Indexer can be divided into two main categories. The first is the actual hardware setup of your Indexer, which is covered by more detailed connection drawings in Chapter 6 - *Hardware Reference*. The second category is the software configuration of the control. This includes distance, acceleration and velocity scaling as well as I/O configuration.

All of the software configuration can be done via the keypad, or via RS-232C using the *Application Developer.* The configuration details presented here are from a keypad user's perspective, via the keypad menu structure and step-by-step keypad instructions.

IDCMotion™ Application Developer and RS-232C users should refer to this chapter for detailed explanations of configuration parameters. Details on using IDC's *Application Developer* can be found in Chapter 5 - *Programming with RS-232C*. For RS-232C terminal users, non-Windows PC users, or PLC users, the equivalent 2-character ASCII configuration commands are detailed in Chapter 5. In this chapter, the 2-character ASCII command appears in brackets next to the keypad command. *IDCMotion™ Application Developer* users will find that the Windows dialog boxes follow the keypad menu structure very closely.

Typical Indexer Application

The 961 and 962 easily interface with a wide variety of stepper and digital servo drives which accept industry standard "Step and Direction" or "CW/CCW Step" control signals. The frequency range of the step output signal allows control of drives, ranging from the simplest full-step motor drive to the highest speed digital brushless servo motors. An block-diagram of an application using IDC's 962 Indexer and keypad is shown below:



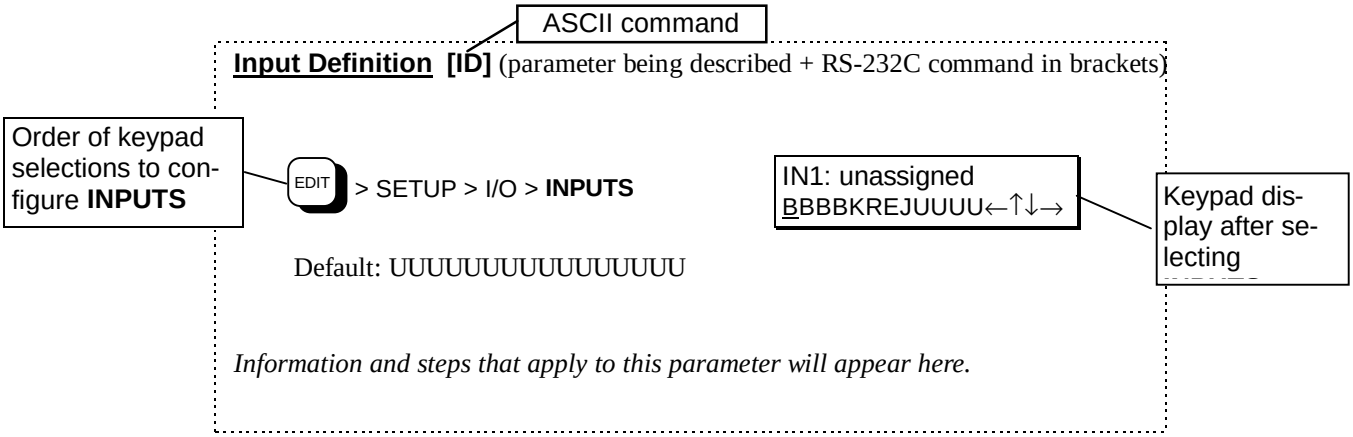
Configuring Your Software

This chapter contains details and directions for customizing the 961 or 962 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 ensure that important parameters are not overlooked. This section is presented from the point of view of the FP220 Keypad user. The directions that follow will take you through each of the SETUP menus in the keypad, and give you details about each of the choices you will be asked to make.

IDCMotion™ Application Developer and RS-232C users should refer to this chapter for detailed explanations of configuration parameters. Details on using IDC's *Application Developer* can be found in Chapter 5 - *Programming with RS-232C*. For RS-232C terminal users, non-Windows PC users, or PLC users, the equivalent 2-character ASCII configuration commands are detailed in Chapter 5. In this chapter, the 2-character ASCII command appears in brackets next to the keypad command. *IDCMotion™ Application Developer* users will find that the Windows dialog boxes follow the keypad menu structure very closely.

The task of configuring your 961 or 962 Indexer 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.

Each SETUP procedure follows the format of the example below:



Using the SETUP Parameters to Configure Your System

Press the EDIT key, then the SETUP (F2) key to reveal nine parameters as shown to the right.

```
- - - -- ↑ SETUP ↓ - - - --
MOTOR   ENC    MECH
```

Press ↓ or ↑ to access the the remaining six parameters.

```
- - - -- ↑ SETUP ↓ - - - -
I/O      JOG      HOME
```

Configuring your system with the keypad begins below:

```
- - -  - ↑ SETUP ↓ - - - -
PROG   RS232    MISC
```

Configuring Your Motor

Configuring the Motor Type [MT]

EDIT > SETUP > MOTOR > **TYPE**

```
--Axis One Motor Type --
←↑      INDEXER      ↓→
```

Default: INDEXER

The selection automatically defaults to INDEXER. There are no other options with the 961 and 962.

1. Use ← and → keys to select an axis.
2. Ensure display reads INDEXER.

Configuring Drive Resolution [MR]

EDIT > SETUP > MOTOR > **D-RES**

```
- Axis One Drive Res -
←↑        25000        ↓→
```

Default: 25000 steps/rev.

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; 1,000; 2,000; 5,000; 8,000; 10,000; 18,000; 25,000; 25,400; and 36,000.

Your application will dictate the choice of D-RES. For example, if you want to move in 0.1 degree increments, a D-RES choice of 18,000 will allow 50 motor steps per degree and prevent any resolution-induced rounding errors. Setting the drive resolution automatically adjusts the pulse-width as shown below.

Drive Resolution in steps/rev	Pulse-Width	Drive Resolution in steps/rev	Pulse-Width
200	23.841 µs	10,000	357 ns
400	23.841 µs	18,000	357 ns
1000	9.536 µs	25,000	357 ns
2000	1.192 µs	25,400	357 ns
5000	1.192 µs	36,000	238 ns
8000	596 ns		

Configuring Motor Direction [MD]

EDIT > SETUP > MOTOR > **DIR**

```
- Axis One Motor Dir- -
←↑    POSITIVE    ↓→
```

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

Configuring Your Encoder

Configuring Encoder Mode [EM]

EDIT > SETUP > ENC > **MODE**

```
- Axis One Enc Mode-
←↑ OPEN LOOP ↓→
```

Default: OPEN LOOP

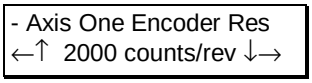
This option sets the encoder mode for each axis. Encoder feedback is strictly optional with the 961/2 Indexers, but if you use an encoder, IDC recommends that you maintain a 4:1 motor-step to encoder-step ratio to prevent end-of-move dither.

1. Use ← and → keys to select an axis.
2. Use the ↑ ↓ and ENTER keys to select the encoder mode:

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. (See Following Error)
CLOSED LOOP	The actual encoder position is displayed on the screen. All subsequent moves are calculated from this actual position. All moves are based on encoder pulses. Stall detection is enabled. Positioning resolution will equal the resolution of your encoder.
SERVO-CLOSED LOOP	Displays actually encoded position, however, moves are based only on commanded OPEN LOOP position. Stalls are detected in this mode.

Configuring Encoder Resolution [ER]

EDIT > SETUP > ENC > **E-RES**



Default: 2,000 counts/rev.

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.

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter the encoder resolution.

Configuring Following Error Limit [FE]

EDIT > SETUP > ENC > **FOL-ERR**

Default: 750 motor steps

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: 0-520,000 motor-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 fault must be cleared before motion can occur. A Stop or Kill, via programmable inputs or RS-232, the ESC key or a RESET will clear a Following Error fault.

Configuring Your Mechanical Parameters

Through the MECH SETUP menu, your Indexer allows you to program 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.

Pressing MECH displays three menu choices:

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

Pressing ↓ reveals three additional menu choices:

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

Configuring the Distance Unit [DU]

EDIT > SETUP > MECH > **DIST**

```
- Axis One Dist Units -
←↑       revs       ↓→
```

Default: 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.)
- Make certain that your Gear Ration (GR) option is set to accurately reflect the Distance Unit.
- If *steps* is chosen, the control automatically fixes the RATIO (see following).

Configuring the Gear Ratio [GR]

EDIT > SETUP > MECH > **RATIO**

```
- - - Axis One Ratio - -
←↑      1  to  1     ↓→
```

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. Make certain that the RATIO accurately represents the Distance Unit (DU).

1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a ratio expressed as two integers. Ex: when entering output shaft revolutions of a 5:1 gearbox, enter "5 to 1" rather than "1 to .2"

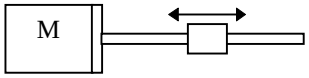
Notes:

- 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 entering 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

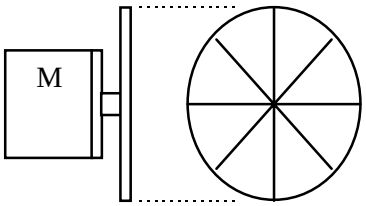
DIST = inch
RATIO = 4 to 1



Units Example - Rotary Index Table

- Desired distance units: 1/8 of a revolution

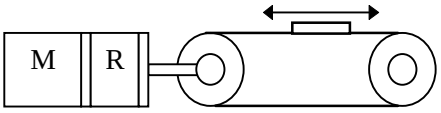
DIST = index
RATIO = 1 to 8



Units Example - Gear Reduced Tangential Drive System

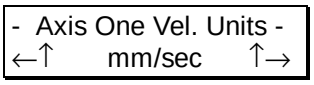
- Desired distance units: mm
- Reducer: 5:1 reduction

Drive pulley: 6 inch circumference
5 revolutions of motor travel results in 152.4 mm of linear load 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



Configuring the Units of Velocity [VU]

EDIT > SETUP > MECH > **VEL**



Default: rps (motor Revolutions Per Second)

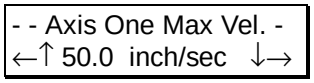
Use this option to select your velocity units. All velocity values specified in the system will be expressed in these units.

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)

Configuring Maximum Velocity [MV]

EDIT > SETUP > MECH > **VMAX**



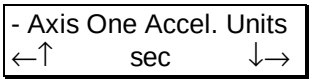
Default: 50 {velocity units}

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.

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

Configuring Acceleration Units [AU]

EDIT > SETUP > MECH > **ACCEL**



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)

Configuring Your Inputs & Outputs (I/O)

Your Indexer 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 thermocouple (Type K or J) inputs. The control is completely protected from damage that might be caused by accidentally interchanging modules. Simply insert the modules and configure each position in the OPTO menu as either an input or an output. If a module is located incorrectly, the control will not respond correctly, but no damage will occur.

Your Indexer 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 controls. Other manufacturers' *analog* opto modules do not fit into a G4 footprint.

For more information on how to use your Indexer's inputs and outputs in an application, refer to Chapter 3. *Programming Your Application*, and Chapter 4. *IDeal Command Reference*.

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

Configuring Input Definition [ID]

EDIT > SETUP > I/O > **INPUTS**

```
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:

Input Characters and Keypad Display

Character	Keypad Display
B	BIN PROGRAM
C	BCD PROGRAM
D	LOCK KEYPAD
E, e	EXTEND JOG 1, EXTEND JOG 2
***F, f**	SET FORCE 1, SET FORCE 2
I	INT (RUN98)
J, j	JOG SPEED 1,JOG SPEED 2
K	KILL
M, m	SHUTDOWN 1, SHUTDOWN 2
N	ANALOG INPUT
P	PAUSE/CONTINUE
R, r	RETRACT JOG 1, RETRACT JOG 2
G	REGISTRATION (1 FOR AXIS 1, 2 FOR AXIS 2
S	STOP
U	UNASSIGNED
V	DATA VALID
W	WARM BOOT

*B8961/2 only

Input Character Descriptions

B Binary Program Select

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

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.

D Lock (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-232C 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.

G Registration

For the Registration (RG) command to function, Inputs #1 and #2 must be configured as Registration inputs. The registration input for axis one is input #1, axis two's is input #2. No other inputs will work. See the RG command for more details.

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 (INT98CTRL) and (ARM INT98) variables. The (INT98CTRL) variable determines whether Interrupts can be disabled 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, every interrupt results in an immediate jump to program 98. Even if you just entered program 98.

This state is backwards compatible with earlier revision 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 re-enable them by setting (ARM INT98)=1. This allows you to control when you want to re-enable Interrupts in your interrupt service routine (program 98).

To summarize, when (INT98CTRL)=1:

If (ARM INT98)=0
Interrupts are ignored. (ARM INT98) is internally set to 0 on the first edge 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 controlling the ability of program 98 to interrupt itself.

If (ARM INT98)=1
The system is awaiting the first 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 the input is ON it will jog at the Jog High speed. If the input is not configured the jog inputs will induce 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: instantaneous deceleration could cause damage to mechanics. The **Stop** input provides a more controlled halt.

M, m Motor Shutdown: (M specifies axis 1, m specifies axis 2)
May be activated when the control is not running a program and the motor is idle. Selecting shutdown (M, m) will disconnect power to the motor, which removes current (torque) and allows the motor to spin freely.

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 Indexer 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 *Chapter 5 - Programming Your Application* for details on using analog I/O.

P Pause/Continue

When 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** and **RG** commands in Chapter 3 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-232C will also clear the fault.

The velocity is determined by the **Jog Speed (J)** input and the Jog Low and Jog 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 in Chapter 3 for more information on stopping moves without halting command execution.

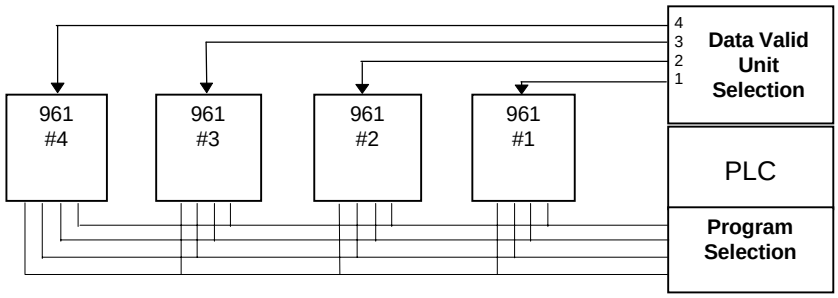
U Unassigned

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 this 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, using the Data Valid input reduced the number of wires by one-half.



W Warm Boot—System Reset

Resets the Indexer, 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.

Configuring Your Outputs

Configuring Output Definition [OD]

EDIT > SETUP > I/O > **OUTPUTS**

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 can not 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 below

Output Characters and Keypad Display

Character	Keypad Display
A	AMP FAULT
B, b	BRAKE 1, BRAKE 2
C	OVER CURRENT
D, d	DIRECTION 1, DIRECTION 2
F	FAULT
H, h	AT HOME 1, AT HOME 2
***K, k**	AT CL LIMIT 1, LIMIT 2
L	LIMIT ERROR
M, m	MOVE DONE 1, MOVE DONE 2
N	ANALOG OUTPUT
P	PROGRAMMABLE
S	STALL
T	TORQUE MODE

* B8961/2 only

Output Character Descriptions

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. Note: This is not an all-inclusive fault output. Use F-Fault for this.

B, b Brake (B specifies axis 1, b specifies axis 2)

CAUTION

IDC offers brakes for either the actuator screw, or as an integral part of the B-series motor. Though both types of brakes are highly effective, there are specific trade-offs that the user should be aware of regarding each type of brake. Please be sure to discuss the issue of brakes with an IDC Applications Engineer or with your distributor.

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. The **Brake** output is normally disengaged, which is actually an ON condition. When a fault occurs, power to the brake is removed and the brake is engaged. This is a "fail-safe" type of brake, controlled by an OPTO module, and it requires a customer supplied, 110VAC power supply, or 24 VDC with B Motors.

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 is commanded in the reverse direction.

F Fault
The fault output acts as an all-inclusive fail-safe output. Under normal operation the output is grounded (ON) and goes high(OFF) when any type of fault occurs. 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 a number of ways:

- Shown on keypad's display
- Over RS-232C using the **SS**, **SD**, and **SA** status commands (see Chapter 7)
- 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.

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.

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/Os may be configured as analog outputs. To use an analog output module the position must be configured as an analog output. This tells the Indexer that the output is no longer a discrete output and ensures that the output signal is sent properly.

Analog signals are set by assigning a value to reserved variables (AO9)-(AO16) corresponding to OPTO positions 9 through 16. See Chapter 3 - *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.

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

T Torque Mode
Not yet implemented.

Configuring Opto Definition [OP]

EDIT > SETUP > I/O > **OPTOS**

```
OPTO14: Output
IIIIOOOO        ←↑↓→
```

Default: IIIIIIII

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 ← and → keys to select an OPTO channel. The top line of the display indicates 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 Indexer 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 manufacturers' *analog* modules do not fit into a G4 footprint.

IDC stocks the following Opto modules, which may be specified when ordering an Indexer:

Order Code (p/n)	I/O Module Description	Opto-22 P/N	Greyhill P/N
A (PCB-1210)	10-32 VAC/VDC Input	G4IDC5	70G-IDC5NP
B (PCB-1211)	TTL Input	G4IDC5K	----------
C (PCB-1212)	35-60V DC Input	G4IDC5G	70G-IDC5G
D (PCB-1213)	90-140 VAC Input	G4IAC5	70G-IAC5
E (PCB-1214)	180-240 VAC Input	G4IAC5A	70G-IAC5A
F (PCB-1215)	5-60 VDC, 3 Amps Output	G4ODC5	70G-ODC5
G (PCB-1216)	12-140 VAC, 3 Amps Output	G4OAC5	70G-OAC5
H (PCB-1217)	Output 24-280 VAC, 3 Amps	G4OAC5A	70G-OAC5A
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		73G-ITCK
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.

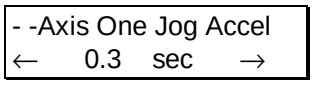
Configuring Your Jog Parameters

Your Indexer's keypad gives the programmer (and the machine operator if desired) a convenient way to jog the motor.

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

Configuring Jog Acceleration [JA]

EDIT > SETUP > JOG > **ACCEL**



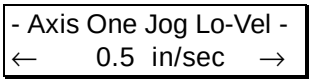
Default: 0.3 {Accel Units}

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.

Configuring Jog Low Velocity [JL]

EDIT > SETUP > JOG > **LO-VEL**



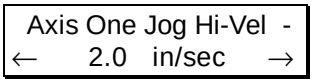
Default: 0.5 {Velocity Units}

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 new low jog velocity value in the same units you selected in the SETUP > MECH > VEL menu.

Configuring Jog High Velocity [JH]

EDIT > SETUP > JOG > **HI-VEL**



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 new high jog velocity value in the same units you selected in the SETUP > MECH > VEL menu.

Configuring Jog Enable [JE]

EDIT > SETUP > JOG > **ENABLE**

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 an operators from accessing them.

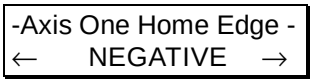
1. Use ← and → keys to select an axis.
2. Use ↑ and ↓ keys to enable and disable the function.

Configuring Your HOME Parameters

Your Indexer 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.

Configuring Home Edge [HE]

EDIT > SETUP > HOME > **EDGE**



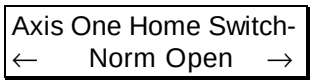
Default: NEGATIVE

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.

Configuring Home Switch [HS]

EDIT SETUP HOME **SWITCH**



Default: Normally Open

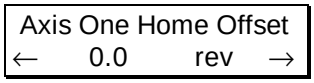
This option select the type of switch used for the home input for each axis. A Normally Open switch connects to ground when activated. A Normally Closed switch is pulled high when activated.

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

Configuring Home Offset [HO]

EDIT > SETUP > HOME > **OFFSET**

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.

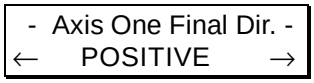
1. Use ← and → keys to select an axis.
2. Use the numeric keys to enter a new home offset value in DIST units.

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.

Configuring Home Final Direction [HF]

EDIT > SETUP > HOME > **FINAL**



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.

Configuring Your PROGRAM SETUP Parameters

Your Indexer is capable of running any pre-defined program on power-up, including inputs configured as BCD or Binary Program Select (usually from a PLC or thumbwheel switches).

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

Configuring Power-Up Program [PU]

EDIT > SETUP > PROG > **PWR-UP**

```
- -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 Indexer 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 numeric keys to enter a program number.

Configuring Scan Conditions [SN]

EDIT > SETUP > PROG > **SCAN**

```
- - -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 Indexer 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.

Configuring Scan Delay [DY]

EDIT > SETUP > PROG > **DELAY**

```
- - - 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 the numeric keys to enter a value in ms.

Note: See Data Valid Input Configuration for an alternate approach.

Configuring Your RS-232C SETUP Parameters

If you plan to use the serial communications port on your Indexer, 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 Indexer.

```
- - RS-232C SETUP - -
ECHO   UNIT#
```

Fixed RS-232C parameters:

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

Configuring Echo Enable [EC]

EDIT > SETUP > RS232 > **ECHO**

```
- - RS-232C Echo - -
↑       ENABLED       ↓
```

Default: ENABLED

This option is used to enable or disable the RS-232C ECHO. If 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 ECHO.

Configuring Unit Number [UN]

EDIT SETUP > RS232 > **UNIT#**

```
 -  - - Unit Number - -  -
NUMBER: _
```

Range: 1-99
Default: 1

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 more information on this type of application. Use the numeric keys to enter the unit address.

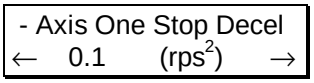
Note: See AA Command in Chapter 5, *Programming with RS-232C*, for Auto-Addressing.

Configuring Your Miscellaneous Setup Parameters

The miscellaneous set-up (MISC SETUP) parameters include auto-formatting of the keypad display, and setting the deceleration rate used with a stop input (or with the ESC key while an axis is moving).

Configuring Stop Decel Rate [SR]

EDIT > SETUP > MISC > **STOP-RATE**



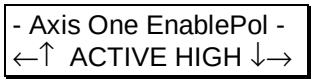
Default: 100 rps^2 (units fixed at motor rps^2)

This option is used to set the deceleration rate for each axis whenever a 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.

Configuring Enable Line Polarity [EL]

EDIT > SETUP > MISC > **ENABLE**



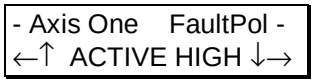
Default: ACTIVE HIGH

This command sets the drive polarity to allow you to enable or disable an amplifier. It can be configured as TTL ACTIVE LOW (GND), or as TTL ACTIVE HIGH (5V).

1. Use the ← → keys to select an axis.
2. Use the ↑ ↓ keys to select ACTIVE HIGH or ACTIVE LOW.

Configuring Fault Line Polarity [FL]

EDIT > SETUP > MISC > **FAULT**



Default: ACTIVE HIGH

This command sets the active sense of the drive fault line. It can be configured as TTL ACTIVE LOW (GND), or as TTL ACTIVE HIGH (5V).

1. Use the ← → keys to select an axis.
2. Use the ↑ ↓ keys to select ACTIVE HIGH or ACTIVE LOW.

Printer - make this a blank page

Chapter 3 - Programming Your Application

The purpose of this chapter is to give the programmer the information necessary to begin developing an application with a 961/962 Indexer. There are also several practical examples that can be copied and modified. Other program examples are available from the *IDCMotion*™ disk set.

Indexer Programming Overview

Before beginning to develop a machine control program with an Indexer, the user must decide how the Indexer fits into the overall machine control hierarchy.
The information in this chapter applies to the following three ways that an Indexer may be used:

1. In a stand-alone mode where the Indexer controls all the Inputs/Outputs and motion.
2. With a PLC, where the PLC runs the machine and calls on the Indexer, via program select lines for motion.
3. In a "hosted" mode, the PC sends serial commands to the Indexer for execution.

The Indexer uses a sequential, interpretive, command processor. This means that commands in a program are executed one at a time, and that one command must be completed before the next command is processed. The following example shows this type of program:

Program: **[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. These state changes of outputs 1 and 2 occur 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 easy to remember and powerful. Command descriptions are available on-line using the HELP key within the editor.

If you need help with basic keypad operation, please refer to Chapter 1 - *Using the Keypad*, and Chapter 2 - *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 the *RS-232C Communication* chapter there is more information on other commands such as controller status and upload/download instructions that cannot be used from within the program of a host computer or PLC.

Command	Description	Command	Description
AC	Acceleration	GS	Gosub
CL**	Current Limit	GZ	Go Home to Z only
CT**	Current hold Time	IF	IF condition
DA	Distance Absolute	IV	Input Variable
DC	Distance to a Change	LP	Loop
DE	Deceleration	MC	Move Continuous
DI	Distance Incremental	MS	Message to Display
EA	Enable Amplifier	ON*	On condition
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
GT	Go to	WT	Wait
		" "	Message to Serial Port

*These commands are not yet available.
** These commands are not available with the Indexer. They are only available in B8961/2 Servo Smart Drives.

Variables and Arithmetic

Variables

The Indexer can use variables in any command instead of a number.

Examples include:

- 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 Indexer 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 parentheses, **(**variable name**)**. Parentheses 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 pre-defined in the control. They can be used throughout your programs in expressions, to set voltages, to test conditions, or even to display information to 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
(POS1), (POS2)	Current Position 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-only
(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 Only
(INT98CTRL)	Enables/disables (ARM INT98) trigger option - Refer to Ch. 2, Configuring Your Inputs, Input Descriptions, Input I.	Read and Write
(ARM INT98)	Enables/disables INT98 input if (INT98CTRL) is enabled	Read and Write

Examples:

- (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 Smart Drive
- (TEMPERATURE)=(AI9) Reads in temperature from an analog input
- (AO15)=4012 Sets the analog output to 4012
- (EE1)=(PIECES) Stores the value of Pieces in the EEPROM variable EE1

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 variable are reset at power down or reset. Every time one of these variables is changed (i.e. used on the left side of a equal (=) sign, the new value is written to, and stored in the user non-volatile EEPROM.

CAUTION

Caution must be used when using these variables. Since EEPROM 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 the POS1 and POS2 variables. The Indexer will allow only 1000 EEPROM writes between power cycles. This limit has been set to prevent a simple programming mistake or misunderstanding from permanently damaging the Indexer'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 EEPROM, so the lifetime of the EEPROM 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 Indexer 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)

You can not enter:

• (X)=1+2-3	This statement is not legal, because it has more than one operand.
• (Length)=(Total)*.03125	The Indexer 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
  (X)=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 Indexer 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 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.

Expressions

The Indexer supports three conditional expressions, less than (<), equal to (=), and greater than (>). 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.

Multi-axis Operation

IDC's IDeal command language is intended to be intuitive, yet powerful enough for normally difficult applications. The Indexers allow you to make multi-axis, synchronized, moves or multiple independent moves that merely 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.

The best way to illustrate this is through 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 they start at the same time, the moves are independent, and will follow the respective accels and velocities, so the axis may not stop moving at the same time.

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

- Start each axis based on different conditions.
  1. Start one axis with an input while another is moving.
  2. Start one or more axis moving based on one input.
  3. Start different axis based on different inputs.
- Reduce the size of some programs by using fewer "move" commands.

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

GO*n*,*n*
n=0-16, integers only
The extended syntax always explicitly defines which axes will move. Only those axis that have an active input defined by **n** will move. The last move parameters used for that axis define the shape and length of the move profile.

An axis will start moving when the input defined by *n* turns on. You can start different axes at different times by using different *n*'s for each axis.

Setting n=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 can save 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.

Different n's for each axis allows you to start one and wait for an input on another, start both based on different inputs, etc. 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

Other Typical Programming Examples

The following example programs will give you an idea of how the IDeal command language can be used solve simple tasks. There are more extensive and elaborate example programs in the DEMOS.idc file that came with your IDCMotion™ disk set. This file can be accessed from application developer.

To aid your program documentation, comments can be placed in brackets. These comments are stripped out of the program as it is downloaded to help conserve memory down 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

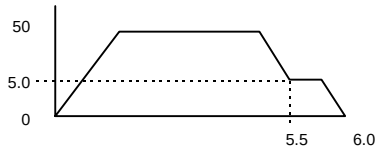
[GET PARTS]	Name the subroutine
MS1,""	Clears the Display
MS1,"How many?: "	Writes string beginning at character 1, top line
IV12,(PIECES)	Waits at 12th character for the # of pieces.
MS1,""	Clears the Display
MS1,"How long?: "	Writes string beginning at character 1, top line
IV12,(LENGTH)	Waits at 12th 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)

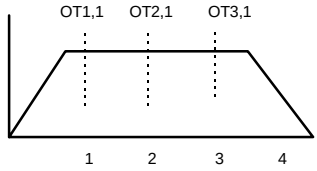
Fast In, Slow Feed Move (Using the Distance to Change (**DC**) command)

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

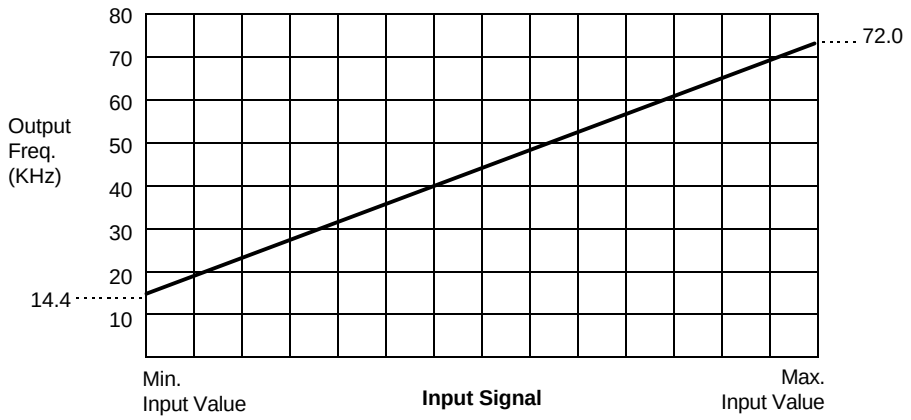
☞ The value of the analog system variables (AI9-16) are scaled from 14,400 to 72,000 Hz. This value is actually a scaled frequency read by the Indexer. These input values are updated every 16 milliseconds . If your program needs to display this value in units like VOLTS, you will need to scale it in 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 the K thermocouples.

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)-14070	Remove frequency offset
(VOLT)=(VOLT)*1.777	Scaling factor mult. by 10,000
(VOLT=(VOLT)*.0001	Scaling back to volts

The variable (VOLT) is now is 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 temperature in from analog input
WT(AI12)<45000 GO	Wait for analog input 2 <45000 (<5.5 VDC using the previous example) before moving
IF(AI12)<45000 GO EB	If analog input 2 < 45000 Go



Configuring an Analog Output

The analog outputs on the Indexers 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 (AO*n*). *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

Chapter 4 - IDeal™ Commands

This chapter defines, in alphabetical order, all the IDeal™ commands that can be used in a program. Most of commands are available from the keypad by pressing F1-F2-F3 and the appropriate number. Some commands like “” and EA must be entered with the Alpha key. More information on entering commands via the keypad is available in the *Keypad Communication* chapter.

Some IDeal™ commands are only supported in RS-232C mode. These commands are listed and defined in the *RS-232C Communications* chapter. Developers who plan to use the keypad or Application Developer will not use these RS-232C-only commands.

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

Command *Command Name* *RS-232C Command Syntax*

DI

Distance Incremental ..... syntax - DI±n, ±n

If the command has a default, it will be listed here.
Units: selected from the EDIT / SETUP / MECH menu
Range: unit scaling dependent

Command Definition

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 usually used in applications where there is no concern for origin, such as feed-to-length applications. They 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 + direction. Move 1 more unit in the + direction. Move 4 units in the - direction.

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

Note: Additional programming examples are found in Chapter 5.

Summary of IDeal™ Commands

Command	Description	Command	Description	Command	Description
AC	Acceleration	GH	Go Home	MS	Message to Display
*CL	Current Limit	GO	Go (Start a Move)	ON	On Condition
*CT	Current Hold Time	GS	Gosub	OT	Output
DA	Distance Absolute	GT	Go To	RG	Registration Move
DC	Distance to a Change	GZ	Go Home to Z Only	SP	Set Position
DE	Deceleration	IF	If Then	SQ	Square Root
DI	Distance Incremental	IV	Input Variable	ST	Stop on Input
EA	Enable Amplifier	LP	Loop	TD	Time Delay
EB	End of Block	MC	Move Continuous	VE	Velocity
EN	End of Routine			WT	Wait
FK	Function Key			“ “	Message to Serial Port

* B8961/2 only

AC

Acceleration..... syntax - ACn,n

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

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:		
	AC2 VE12 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.

DA

Distance Absolute .....syntax - DA±n, ±n

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent

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 - DCn,n

Units: selected from the EDIT > SETUP > MECH menu.
Range: unit scaling dependent

DC is used to define complex, multiple velocity move profiles, or to change an Output at a specific point during the move. It defines the distance at which a change will occur, "on the fly", while the motor is still moving. At the specified distance you can change the velocity, acceleration, deceleration or change the state of an output(s).

The DC command must follow the 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 should 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.

Examples:
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 DC3 VE15 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 *MORE* units of motion, turn off output 1 and turn on output 2.

AC.05 DE.15 VE50 DI15 DC5 VE10 DC5 VE5 GO
At a starting speed of 50, begin moving an incremental distance of 15 units. After 5 units, ramp down to 10 speed. After 5 *MORE* distance units ramp down to 5 speed and continue until the final position is reached.

AC.1,.1 VE10,20 DI10,10 DC5,5 VE20,10 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 - DEn,n

Units: sec, rps^2 or unit/s^2 (selected from the EDIT > SETUP > MECH menu)
Range: unit scaling dependent
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.5 VE12 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±n, ±n

Units: selected from the EDIT > SETUP > MECH menu
Range: unit scaling dependent
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 usually used in applications where there is no concern for origin, such as feed-to-length applications. They 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 - EAn,n

Range: 0 (disable), 1 (enable)
EA sets the state of the amplifier enable signal. The polarity can be changed in EDIT > SETUP > MISC > ENABLE.

Example: EA0,0 Disables the amplifiers on axis one and two.

EB

End of Block .....syntax - EB

The EB command designates the **E**nd of a **B**lock of loop or IF commands. Every LP 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

Syntax: EN

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 that called the subroutine. If the routine was not called from another program, the EN command simply stops 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 - FK1,2,..,28

Range: 1-28

Note: 24, the ESC, key cannot be assigned, since it stops a program

The FK command allows you to redefine what the keypad key presses mean from within your program. The FK command pauses processing until any of the buttons you have "armed" are pressed. The number of the button pushed 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's 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 chart for the value of (FKEY) returned for each key.

	F1 = 1	**F2** = 2	**F3** = 3	
RUN = 4	**EDIT** = 5	**HELP** = 6	**COPY** = 7	**DEL** = 8
1 = 9	**2** = 10	**3** = 11	←= 12	→= 13
4 = 14	**5** = 15	**6** = 16	↑= 17	↓= 18
7 = 19	**8** = 20	**9** = 21	**.** = 22	**,** = 23
Cannot be used	**0** = 25	**+/-** = 26	**ALPHA** = 27	**ENTER** = 28

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±n,±n

Range: unit scaling dependent
Units: velocity units selected in EDIT > SETUP > MECH menu
Direction: positive (+) direction established in EDIT > SETUP > MOTOR menu

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 can not 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 which side of the home switch this final approach will be based from. In a "closed loop" mode Go Home routine, the control will additionally slow to a creep speed and stop when it sees the encoder's "Z" Marker Pulse after seeing the reference edge of the switch. If a marker pulse is not seen within one motor revolution after the reference edge of the switch is seen, the final homing routine will be aborted.

Examples:

AC.5 DE.5 GH-20	Go Home in the negative direction at a speed of 20
AC.5 DE.5 GH20,18	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.

GI

Go Immediate.....To be implemented in a future version of software**.**

GO

Go (Start a Move).....syntax - GO or GOn,n

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.

GOn pre-calculates the move and waits for Input number "n" 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 (WTn GO) yet it pre-calculates the move. Like other commands using I/O, GOn does not restrict you from using an input even if it has been configured for some predefined function.
Range: n =1-16

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.

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.

GS

Gosub.....syntax - GS and GS[name]

Range: n = 1-199, [name] = any legal program name
(n=1-400 w/-30k memory option)

Jump to program number or name, 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 *standard* firmware 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 - GTn or GT[name]

Range: n = 1-199, [name] = any legal program name
(n=1-400 w/-30k memory option)

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 is never run. Use the GS command if you want to return to this program and goto program 30.

GZ

Home to Z Channel..... syntax - GZ±

This command will rotate the motor in the direction specified by ± until the encoder Z pulse is encountered. This command does not require home switches and is sometimes used in rotary applications to eliminate the need for a home switch.

The GZ command will home very slowly to ensure that the Z pulse is not missed.

IF

If.....syntax - IFn,xx

Syntax(s): IFn,xx...
IFxx... (assumes first input is input 1)
IF(mathematical expression)

Range: n = starting input number, 1-16
x = 0; input high
x = 1; input low (grounded)
x = *anything else*; don't care input level
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 Smart Drive 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 like "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(AI9)< 55000 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 - IVn,(variable),*min,max*

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

This command allows an operator to input variable information under program control. It is usually used along 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 n. 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 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 - LPn

Default: 0
LP will cause all commands between LP and EB to be repeated "n" times. If LP is entered without a number following it or a 0, the loop will repeat continuously

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

The *standard* 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.

MC

Move Continuous ..... syntax - MC+,+

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

Each Move Continuous segment must contain a GO command. Accelerations, Velocities, and Decelerations 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 moving in 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 deceleration rate) and change directions to the new speed. This makes changing directions based on analog 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 a Mode Continuous 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 do things like:

- 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 or 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 (limited bandwidth, now 10Hz, future firmware will make 100 Hz)
- Make a one or two axis joystick using analog inputs
- Start a continuous move on one axis, and make distance based moves on another

Move Continuous moves will not stop when a program completes. This allows you to write programs that:

- 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.

See examples of programs on following pages:
Example 1: Basic Move Continuous syntax. Demonstrates how to change speed and stop Mode Continuous moves based on time delays and input conditions.

```
MC+             Enable Move Continuous on axis 1
```

AC.1 DE.2	Set the acceleration and deceleration rates
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 at the rate specified by the DE command
WT111	Wait for inputs 1,2, and 3 to go active
VE0 GO	Stop the move at the rate specified by the DE command

Example 2: Demonstrates how to prompt an operator for speed changes on a one axis Indexer . 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 inputted 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 Smart Drive.

[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

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

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

MS allows you to put messages on the keypad's display. Messages are usually 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 all 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.

Example:

```
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 row).

OT

Output ..... syntax - see below

Syntax(s): OTn,xx...
OTxx...(assumes first output is # 1)
Range: n = 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 unchagned
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.
See *Programming Your Application* for information on using analog Opto outputs.

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:

OT14,1	Turn on Output 14, Opto I/O #5
OT12,0D1	Turn Outputs 12 off, leave 13 as is, and turn 14 on
OT110	Turn Outputs 1 and 2 on, and 3 off

“”

Quote..... syntax -“*Any ASCII character*”

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*.

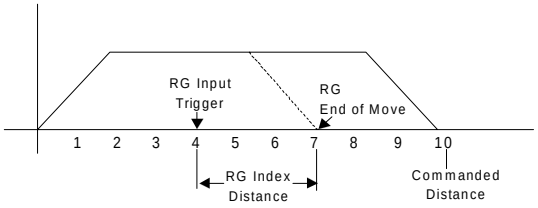
RG

Registration ..... syntax - RGn, or RG,n

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 RG3 DA10 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 what the registration move would be.



Accompanying the programmable Registration Command is the configurable Registration Input: G (also G in RS-232C 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).

Note: Registration Input is only configurable on Input #1 for Axis #1, and on Input 2 for Axis 2.

System Performance when Using the RG (Registration) Command

There is a Capture Delay (reaction time) associated with the RG Command, which is a function of move velocity 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 version hardware of your Indexer this capture delay will either be 164 µs or 5 µs.

The Registration Command is only available on firmware version 3.42 of higher. If you have FPGA version 5.9 or earlier, the delay will be 164 µs. If you have FPGA 6.7 or higher, the position is captured in hardware, and the only delay is the input's opto-isolator (5 µs). Use the keypad's HELP key to determine these versions. Regardless of version, we have found both to be extremely repeatable. However, the version may effect your program slightly. For example, a motor traveling 240,000 steps/sec (30 rps with 8000 step drive resolution) has a capture window of 39 steps at 164 µs and 1 step at 5 µs.

SP

Set Position.....syntax - SP±n,n

Default: 0
Range: Varies based upon Distance Units
Units: selected in EDIT > SETUP > MECH menu
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 - SQn,(var)

Range: 0.0001 to 214748.3645

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 n 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

Range: n = 0-16
ST stops Move Execution upon activating the input specified by n.
ST0,0 Disables/turns off the STn command.

After this command is executed, the specified input is monitored during every "move profile". If the input is activated, the current "move in progress" is terminated, stopping all motion until the input is deactivated or a ST0 is processed. The unit will process and calculate commands, but it will wait at the next GO command until the ST input changes. Immediately following deactivation (release) of the input, the next move profile is executed.

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:
ST1 AC1 DE1 VE25 DA6 GO VE50 DA0 GO EN

Move to absolute position 6 units. If input 1 is activated while moving, Stop Motion. When the input is deactivated, immediately execute the next move profile which is to move to the absolute zero position. If input 1 is not activated the motor would complete its 6 unit move before executing the move back to absolute zero.

TD

Time Delay .....syntax - TDn

Range: .01 to 99999.99 seconds

Delay n 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 *Programming Your Application*.

VE

Velocity .....syntax - VEn,n

Default: 1 motor rev per sec (rps)
Units: selected in EDIT / SETUP / MECH menu
Range: Varies with Velocity Units.

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

Syntax(s): WTn,xx...
WTxx... (assumes first input is input 1)
WT(AIn), expression
Range: n = 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.

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:

WT14,1 GO	Wait for input 14 to equal 1 before moving
WT12,010 GO	Wait for inputs 12-14 to equal 010 before moving
WT110 GO	Wait for inputs 1-3 to equal 110 before moving
WT(AI9)< 45000 GO	Wait for analog input 9 < 45000 before moving

Chapter 5 - Programming with RS-232C

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 Smart Drive controls. IDC provides, and strongly recommends using our Windows-based software tools 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, included in the IDCMotion™ disk set, provides a graphical control configuration environment, a program development editor, and a terminal communication package. *Application Developer's* also provides application upload and download utilities, and an on-line software reference help utility.

Servo Tuner, also included in the IDCMotion™ disk set, is a servo configuration and tuning utility for the B8000 Series Smart Drives. It allows you to select and download motor and tuning parameters, and to create a stimulus for quantitative servo tuning for optimum system response. Servo Tuner is not used with the 961 or 962 indexers.

This chapter is divided into 4 main sections. The first section covers common RS-232C details including baud rate settings and daisy chaining information. All RS-232C users will need this information. The second section covers the installation of the IDCMotion disk. The third covers the details for using IDC's *Application Developer* to setup and program Indexer systems. The fourth section provides details on all 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 Indexer 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.

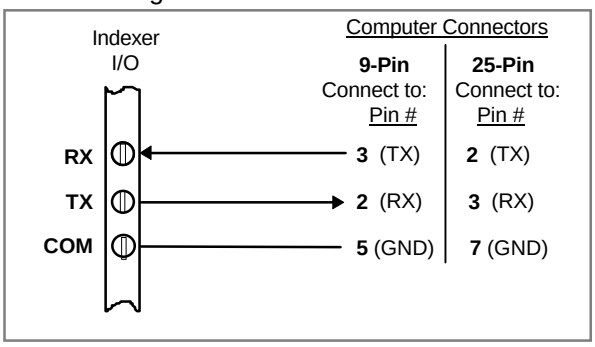
Section 1: RS-232C Protocol

IDC's Indexers 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 user's responsibility to disable this handshaking via software or hardware.

Comm Port Settings:

Baud Rate:	9600
Stop Bits:	1
Data Bits:	8
Parity:	none
XON/XOFF:	yes

Connecting RS-232C to the Indexer:



Troubleshooting Note: If communications between computer and Indexer are unsuccessful, try switching the RX and TX connections on the Indexer. Also, make certain you have chosen the correct Comm Port setting in Application Developer > Comm Port Setup. This setting must match the serial port used on the computer.

Daisy Chaining

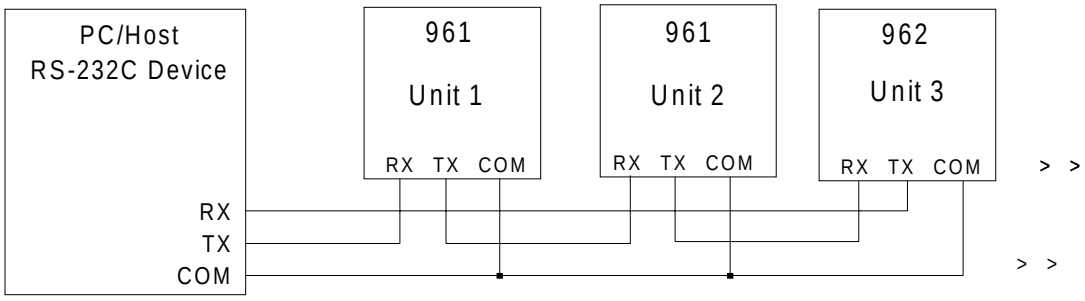
IDC's Indexers also support daisy chaining. The unit address (range 1-99) can be set via the keypad, through Application Developer, or with a terminal program using the Unit Number (UN) command.

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 to work properly.
2. The unit addresses are not required to be numerically sequential, but must be in ascending order. Example: 1, 2, 3, 4, 5 or 10, 20, 25, 29, 31 are valid addressing. 6, 3, 10, 8, 2 is not valid.
3. Do not duplicate unit numbers or addresses.
4. 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.
5. Any loose RS-232C connections or miswiring along the daisy chain will cause communication to fail. Please double check wiring if communication problems arise.
6. "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. Omitting the unit number will cause the command to be seen and executed by every unit on the daisy chain.
7. Firmware version 3.30 or higher, and Application Developer 1.43 or higher are recommended when operating Indexers in a daisy chain configuration.

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: *IDCMotion™* Software

All of IDC's Indexers come with an *IDCMotion™* disk set., *IDCMotion* is comprised of *Application Developer*, *Servo Tuner*, motor data files and demo programs. The programs and data files are automatically installed with a setup utility included on the disk. The *IDCMotion* disk also includes a README file that contains the most up to date information on hardware and software features. The readme file also contains a program listing of demo program included with *Application Developer*.

Installing *IDCMotion™* in Windows version 3.0 or later

With Windows running,

1. Insert *IDCMotion* 5.1 Disk 1 of 2 in a floppy disk drive.
2. In the File menu, click Run.
3. 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:

   a:\setup

4. Click OK.

In Windows 95, click on Start, then click Run. Follow instructions in steps 3 and 4.

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. Please note that this Setup program will over-write any existing versions of Application Developer and Servo Tuner. If you want to save the older versions of these programs, install the new versions to a different directory, or rename the older files.

When you reach the window that says “Click the type of Setup you prefer, then click next”, select **Typical** unless a partial installation is required. Selecting Custom allows you to select/de-select applications and various IDCMotion support files. Application Developer and Servo Tuner may not run properly if these support files are not installed.

The Setup program will then prompt you to select a program group (or program folder in Win’95) to install the program icons. The Setup program will also create a series of sub-directories containing Application Developer, Servo Tuner, and an icon list. Also included in the Application Developer directory is a readme.txt file, a PowerPoint training presentation, and a terminal program for use in troubleshooting daisy-chains.

Common Installation Errors and Remedies

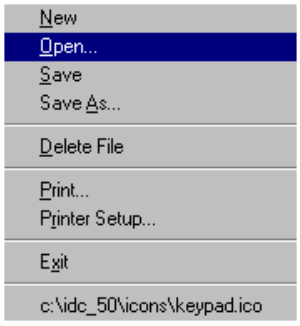
IDC recommends closing all applications, including MS-Mail and MS-Office, before running **SETUP** to help avoid file conflicts. There are no known installation problems with IDCMotion 5.1. Please call the IDC Applications Department at (800) 747-0064 or (415) 382-4300 (from outside U.S.) if you need assistance installing this software.

Section 3: Application Developer™

Application Developer’s graphical environment helps you set-up and program your Indexer 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 the *Configuring Your System* chapter of this manual. Applications (programs and configuration files) may be created, saved, edited and downloaded (sent) to your control(s). *Application Developer* will also upload (receive) a control’s entire setup and program memory.

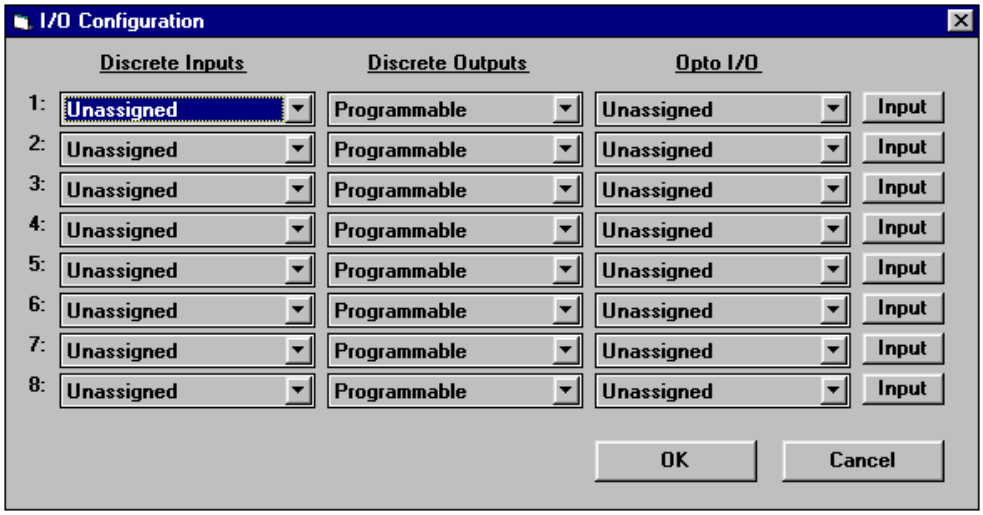
File

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.



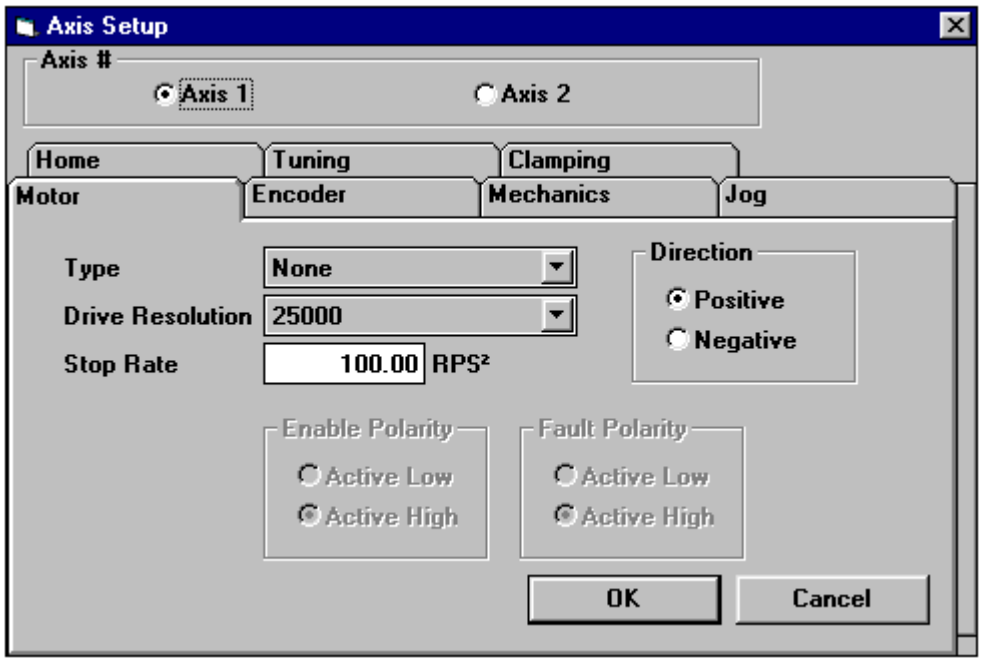
Setup / I/O

Select “I/O” to define a dedicated function each Input and Output. Scroll through the pull down lists and select from available choices. For OPTO I/O click on the option button to the right of the list to select whether the position will be an input or an output. The OPTO I/O pull down function list options will change accordingly. All inputs and outputs, even dedicated function ones, can still be used in program flow control statements like IFs and WTs. More information on these dedicated functions can be found in the *Configuring Your Control* chapter.



Setup / Axes

Use the *Setup / Axes* screen to configure motion related parameters (i.e. unit scaling, jog speed and acceleration, homing, encoder resolution, etc.).

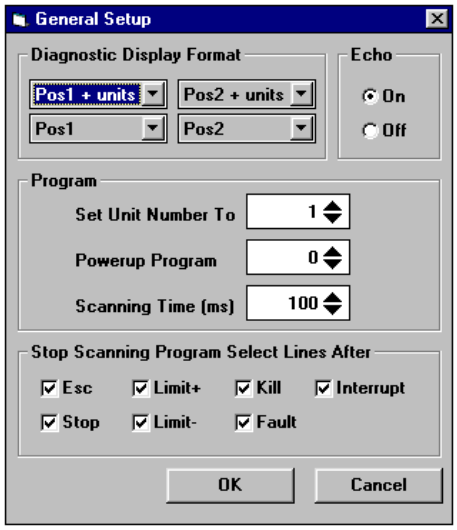


Complete all the Tabs available for Axis 1 before moving on to Axis 2. After all six tabs have been completed, select the Axis 2 radio button and repeat the process. These dialog boxes follow the same order as the process described in the keypad configuration. It is

recommended that users follow the procedure described in *Configuring Your System* while filling in the dialog boxes.

Setup: General

The *Setup: General* dialog box has four configuration categories has the same choices as the SETUP/MISC menu detailed in the *Configuring Your System* chapter.



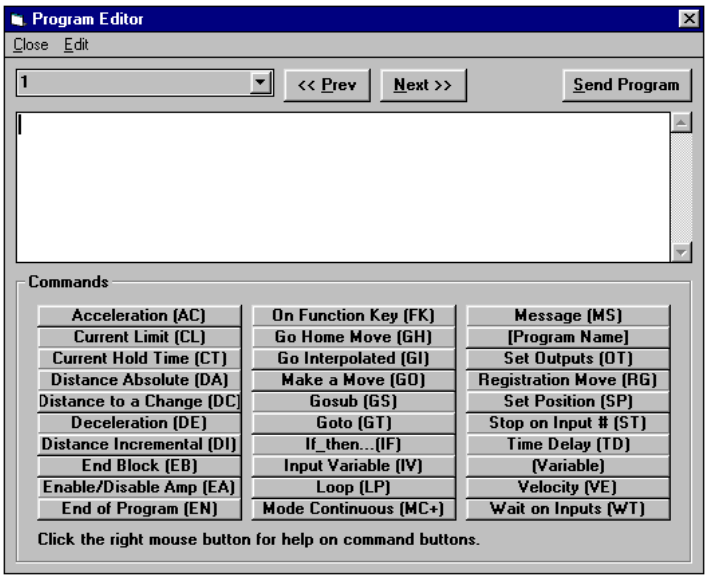
1) Diagnostic Display Format - Only for use with a keypad. *(to be implemented in a future version of software.)*
2) RS-232 echo - Turns the echo on and off.
3) Program- 3 sub-categories
   - Sets the Smart Drive's daisy chain address
   - The program to run on power-up. No program will run if set to 0.
   - The debounce (scan time) time of the program select inputs.
4) Program interruption and execution control. Defines the conditions under which program scanning stops

As always, complete details are in the *Configuring Your Control* chapter.

Setup: Edit Programs

The *Application Developer* Program Editor features the standard Window's Cut, Copy, and Paste functionality, automatic command insertion, and an online manual via F1 or the Right mouse button.

The drop down menu box in the upper left hand corner shows the number and name of the program currently being worked on. This is also where new programs are selected. When the entire file is downloaded to the Smart Drive control, these program numbers correspond to the program numbers the controller uses for binary and BCD program selections.



Use the Grow Editor button to increase the working text area, and eliminate the command prompts. Program comments are placed between brackets {comments}. These comments are not downloaded to the Smart Drive. Total program length, not including comments, is limited to 1k. Total program length with comments is 8k.

Communications:

Send Program

The *Send Program* menu choice downloads the application you have developed. In addition to your motion programs, your application file will include the setup commands derived from the choices you made in the Setup dialog boxes. This *Send* completely configures the Smart Drive 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 Smart Drive. IDC recommends saving the commented version of your application before downloading.

Retrieve Program

The *Retrieve Program* menu uploads the entire contents of a Smart Drive control to a new file that can then be edited, downloaded to another Smart Drive, or saved to a PC file for documentation purposes. This file contains the complete contents of the Smart Drive 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 Smart Drive.

Change Unit Number

Unit Number is used to set the device address of the Smart Drive 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) *Send Program* only sends information to the unit selected here. A new unit number must be set to download to the next unit on the daisy chain. See *RS-232C Protocol*, earlier in this chapter, for hardware information on daisy-chain wiring.

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 Smart Drive control. It is very useful for troubleshooting interactive host/control communications.

Run: Program

Run: Program is used to run a specific program from *App Developer*. Programs can also be initiated via dedicated program select inputs, through the keypad, or via any terminal using the RN command.

Section 4: RS-232C *IDeal™* Command Reference

Overview

Though we strongly recommend taking advantage of the capabilities and convenience of *Application Developer* and *Servo Tuner*, you can configure, program, and run an Indexer 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.). **Note**: Upon request, IDC will share any VisualBASIC™ or other source code we have developed to communicate with our controls via RS-232C. Please be aware however, that IDC is currently unable to provide any level of product support for the use of our proprietary source code.

RS-232C/Indexer operation is divided into four categories of RS-232C commands. The first category is ***Setup Commands***. These are the commands that IDC's *Application Developer* program uses to configure the Indexer according to the choices made in the SETUP dialog boxes. These ***Setup Commands*** include the syntax of the command, but the full command definitions and examples are found in Chapter 2 - Configuring Your System.

The second category is **RS-232C Syntax for *IDeal™* Commands**. These are commands that can be executed immediately over RS-232C or downloaded via RS-232C to the Indexer'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-232CC commands are listed in the RS-232 Command Reference, but the full definition and examples are listed in Chapter 4 - *IDeal*™ Command Reference.

The third category of RS-232C commands is ***Status Commands***. These commands bypass the normal command buffer and are executed immediately, regardless of what else the Indexer has been asked to do. This category includes commands such as instantaneous position reporting, drive status, and emergency kill and stop commands.

The final category is ***Interface 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 Indexer.

Sample RS-232C File

To familiarize yourself with *IDeal™* RS-232C commands, review the following example of a typical file created by *Application Developer* for Download to a S6961/2 controller. You will need to generate a similar file to configure your Smart Drive. Individual programs can be downloaded with this configuration file, or separately at runtime. The Smart Drives can also execute run time commands in an "immediate" mode outside of any program. More details on this mode of operation are in the *Command Syntax* section of this chapter.

```
{ IDC Motion Application Developer }
{ c:\idc\manual.idc, 05-28-1995, 22:18:49 }

{ Unit# and begin download }
01LA

{Setup Parameters}
MT:4,10,10
MR:5,8,8
AU:2,4,4
VU:0,0,0
DU:2,0,0
GR:25:10,1:1,1:1
ID:MmEeRrKRNNSS----
OD:MmHhPPPP----PPPP
OP:IIIIOOOO
JA:0.3,.3,.3
JL:0.5,.5,.5
JH:2,2,2
JE:0,0,0
HF:0,1
HL:0,0
HO:0,0
PU:0
SN:1111111
DY:100
EM:2,0
ER:8000,2000
FE:4000,25000
BK:0,0,0
DF:4,5,1,2
SR:1,1,1
MD:0,0,0
MV:80.00,80.00,80.00
AH:0,0,0
AW:0,0,0
FA:0,0
FV:0,0
KI:50,0
KP:14049,0
KV:15000,0
{Programs}
PR:1 [MANUAL EXAMPLE] AC4 DE4 VE7.5 DA10 GO EN EP
PR:4 [EXAMPLE 2]  MS2,"PRESS F1 TO PROCEED"  FK1  GT1  EN EP

{End download}
EX
{Global parameters}
EC:1
```

Setup Parameters are only downloaded once. They are stored in the non-volatile memory of the Smart Drive. Setup commands cannot be stored or executed in a run-time program.

Executable programs. Can be called via RS-232, run on power-up or select via digital program select inputs.

Command Syntax

All *IDeal™* RS-232C commands use two letter upper case ASCII characters. Command delimiters can be a carriage return (<cr>) or space (<sp>) character.

The RS-232C command that follow define IDC's command syntax. A brief command description is given here, but the full command definition is found in Chapter 4. The following listing is intended only to help non-Windows, RS-232 users with command syntax. SETUP commands are defined in *Chapter 4*, and the *IDeal*™ programming commands are defined in Chapter 6.

The table below describes the abbreviations and format used in our command syntax definitions:

For example: <n>**AUi,i**.

Letter or Symbol	Description
<n>	Optional unit address number, the command is sent to all units if no address is specified.
,	Represents a field separator. A comma is often used to separate a value for axis two from axis one.
a	Represents an alphabetic character.
h	Represents a hexadecimal number.
i	Represents an decimal integer number.
r	Represents a 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.

Some *IDeal™* RS-232C 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.

RS-232C Commands

Setup Commands

The quickest way to configure your application is by using the keypad or our *IDCMotion Application Developer* software. Full descriptions of the Setup Commands are in Chapter 2 - *Configuring Your System*. The page you should reference in *Configuring Your System* is listed in the far right column of the command listing.

Command	Name	RS-232C Syntax

AU **Acceleration Units** **nAUi,i**

	Units
i=0	units/s^2 (where "units" is a string defined by the DU command)
1	rps^2
2	seconds

Example: AC0,1 (units/s^2 on axis one, and rps^2 for axis two)

DF **Display Format** **nDFi,i,i,i**

To be implemented in a future version of software.

DU **Distance Unit Label** **nDUi,i**

	Units		
i=0	step (fixs GR @ 1:1)		
1	rev	9	deg
2	inch	10	radian
3	mil	11	grad
4	meter	12	arcsec
5	cm	13	arcmin
6	mm	14	%
7	yard	15	Index
8	foot		

Example: DU3,4 (Axis one units in .001 inch increments, axis two in meters.)
See Also: GR

Command	Name	RS-232C Syntax

DY **Scan Delay** **nDYi**

Example: DY500 (500 ms) Default is 100ms
.

EC **RS-232 Echo Enable/Disable** **nECi**

0 = echo Disabled, 1 = echo Enabled.
Example: EC0 (echo off)
The RS-232 Echo must be enabled for daisy chain operation.

EL **Enable Line Polarity** **ELn,n**

0= ACTIVE LOW, 1= ACTIVE HIGH.

EM **Encoder Mode** **nEMi,i**

0 = Open loop, 1 = Open loop with stall detect, 2 = Closed loop,
3=Servo closed loop (only available with servos, call IDC for details)
Example: EM2,2 (both axes closed loop)

ER **Encoder Resolution** **nERi,i**

Example: ER8000,8000 (axis one 4000 counts/rev, axis two 8000 counts/rev).
The encoder resolution is fixed at 8000 counts/rev when using IDC supplied servo motors.

FE **Following Error Limit** **nFEi,i**

Example: FE1000,100 (axis one 1000 counts, axis two 100 counts).

FL **Fault Line Polarity** **FLn,n**

0= ACTIVE LOW, 1= ACTIVE HIGH.

GR **Units Ratio** **nGRi:i,i:i**

Example: GR4:1,25:762 (axis one 4 motor revolutions per distance (DU) unit, axis two 25 motor revolutions per 762 distance units).

Command	Name	RS-232C Syntax

HE **Home Edge** **nHEi,i**

0 = Positive Edge, 1 = Negative Edge.
Example: HE0,1 (axis one positive, axis two negative).

HF **Home Final Direction** **nHFi,i**

0 = Negative direction, 1 = Positive direction .
Example: HF1,0 (axis one positive, axis two negative).

HO **Home Offset** **nHO±r,±r**

Example: HO1.0,-2.5 (axis one 1.0 distance units, axis two -2.5 distance units)

HS **Home Switch** **nHSi,i**

0 = Normally Closed, 1 = Normally Open
Example: HS1,1 (axis one and two use a NO home switch, this is the default setting).

ID **Input Definition** **nIDaaaaaaaaaaaaaaaa**

Example: IDUUUUUUUU-------- The first 8 inputs are unassigned, and the 8 OPTO positions are configured as outputs. All 16 input states must be specified.
See also: OD and OP
Note: The G (Registration) Command is only valid for Inputs 1 and 2. Input 1 is configurable only for axis 1, and input 2 only for axis 2 when using the G Command.

JA **Jog Acceleration** **nJAr,r**

Example: JA.01,400 (Axis one .01, axis two 400. Units selected by AU command.)

JE **Jog Enable** **nJEi,i**

0 = Jog Disabled, 1 = Jog Enabled
Example: JE1,1 (axis one enabled, axis two enabled).

JH **Jog High Velocity** **nJHr,r**

Example: JH5.0,6.0 (Axis one 5, axis two 6, in units selected by VU command.)

JL **Jog Low Velocity** **nJLr,r**

Example: JL1.5,2 (Axis one 1.5, axis two 2, in units selected by VU command.)

Command	Name	RS-232C Syntax

MD **Motor Direction Reference** **nMDi,i**

0 = Positive direction, 1 = Negative direction
Example: MD0,1 (axis one positive, axis two negative)

MR **Motor Resolution** **nMRi,I**

	Resolution		
i=0	200	6	10000
1	400	7	18000
2	1000	8	25000
3	2000	9	25400
4	5000	10	36000
5	8000		

Example: MR8,9 (axis one 25000, axis two 25400 counts per revolution).

MT **Motor Type** **nMTi,i**

	Model or Type		
i=0	None	6	B41 @ 220V
1	B23 @ 110V	7	H3
2	B23 @ 220V	8	H4
3	B32 @ 110V	9	Other
4	B32 @ 220V	10	Stepper
5	B41 @ 110V	11	Indexer

Example: MT1,3

Command	Name	RS-232C Syntax

MV **Maximum velocity** **nMVr,r**

Example: MV50.0,1.65 (Axis one 50, axis two 1.65, in units selected by VU command.)

OD **Output Definition** **nOD:aaaaaaaaaaaaaaaa**

Example: OD:PPPPPPPP- - - - PPPP All 8 outputs defined as Programmable outputs, and the last 4 OPTO(13-16) positions defined as Programmable outputs. OPTO positons 9-12 are configured as inputs. All 16 output states must be specified.
See also: OP and ID

OP **OPTO Configuration** **nOP:aaaaaaaa**

I = Input, O = Output.
Example: OP:IIIIOOOO (first 4 configured as inputs, last 4 configured as outputs)
See also: ID and OD

☞ OP must be set before ID and OD to prevent ID and OD settings from changing when OP is set.

PU **Power-Up Program** **nPUi**

Example: PU105 (Runs program number 105 on power-up.)

SN **Scan Conditions** **nSNaaaaaaa**

Conditions stopping program select line scanning are represented by each "a" from left to right:

	ESC	Stop	Limit +	Limit -	Kill	Fault	Interrupt
SN	**a**	**a**	**a**	**a**	**a**	**a**	**a**

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).

SR **Stop Deceleration Rate** **nSRr,r**

Example: SR100,50 (Axis one 100 rps^2, axis two 50 rps^2)
Note: Stop Deceleration Rate units are always in rps^2 and are not user selected.

UN **Unit Number** **nUNi**

Example: UN5 (sets unit address to 5).

VU **Velocity Units** **nVUi,i**

i	velocity units
0	units/sec (where "units" is a string defined by the DU command)
1	units/min (where "units" is a string defined by the DU command)
2	rps
3	rpm

Example: VU2,3 (axis one set to rps, axis two set to rpm).

RS-232C Syntax for *IDeal*™ Commands

The first list of commands in this category may be sent to the Smart Drive'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 Smart Drive's volatile memory. For a complete definition of RS-232C Programming commands, see Chapter 4, *IDeal*™ *Command Reference.*

Command	Name	RS-232C Syntax

AC **Acceleration** **nACr,r**

Example: AC10,15

DA **Distance Absolute** **nDA±r, ±r**

Example: DA15,0

DC **Distance to a Change** **nDCr,r**

Example: DC20,20

DE **Deceleration** **nDEr,r**

Example: DE.2,.15

DI **Distance Incremental** **nDI±r, ±r**

Example: EA0,0 disables axis 1 and axis 2

EA **Enable Axis** **nEAi**

i=0 (drive disabled), i=1 (drive enabled)
Example: EA0,0 disables axis 1 and axis 2.

GH **Go Home** **nGH±r, ±r**

Example: GH10,5

GO **Begin Move(s)** **nGO, nGOi,I**

MC **Mode Continuous** **nMC+,+**

OT **Set Outputs** **nOTi,iii... nOTiii...**

Example: OT5,101 (turns on outputs 5 and 7, turns off output 6)

" " **Send String Over RS-232C** **n"user text"**

Example: GO "End of Move" (sends "End of Move" out the serial port after the move)

Command	Name	RS-232C Syntax

OT **Set Outputs** **nOTi,iii... nOTiii...**

Example: OT5,101 (turns on outputs 5 and 7, turns off output 6)

" " **Send String Over RS-232C** **n"user text"**

Example: GO "End of Move" (sends "End of Move" out the serial port after the move)

RG **Registration** **RGn**

SP **Set Position** **nSPr,r**

Example: SP15.0,0 (set axis one position to 15.0, and axis two to 0, in user units).

SQ **Square Root** **SQn,(var)**

ST **Stop on Input** **nSTi**

Example: ST4 DI50 GO (motor will decelerate to a stop or remain stopped if input 4 is activated). ST0 disables the input.

TD **Time Delay** **nTDi**

Example: OT1,1 TD.12 OT1,0 Turn Output 1 on for 120 msec.

VE **Velocity** **nVEi,i**

Example: VE50 (sets speed for the next move to 50, in units set by the VU command).

WT **Wait On Input Condition** **nWTi,ii... nWTiii... nWT(Ali),**

Example: WT1,0 GO (wait for input 1 to turn off before starting move)

The following set of commands can only be executed if they are part of, or within, a program:

Command	Full Name
EB	**End Block**
GS	**Go Sub**
GT	**Go To**
IF	**If**
LP	**Loop**
IV	**Input Variable**
MS	**Display Message**

Status Commands

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 Status Commands

The **Status** commands are provided for two purposes. One is to allow a host control to query the Indexer , in real time, for program, 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 Indexer and generate a return, regardless of what else the Indexer is doing. They do not affect operation of the Indexer, so moves, messages, and output status will change.

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 Indexer by this host device. The following commands are provided so that the host can verify the status of the Indexer 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 immediate commands are also an invaluable system troubleshooting aid. Since they are immediate commands, they will generate a response from the Indexer even if it is 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 Indexer is doing. If a fault is indicated, the Drive and Axis Status commands can give detailed, axis specific, information.

Summary of Status Commands

Status Command	Syntax	Purpose
Input Status	IS	Real time status of discrete and OPTO inputs
Kill	K	Issues immediate halt to current and programmed motion
Model Number	MNn,n	Returns unit model number over RS-232C
Output Status	OS	Real time status of discrete and OPTO outputs
Current Position Axis 1	PA1	Real time position, in user units, of axis 1
Current Position Axis 2	PA2	Real time position, in user units, of axis 2
Stop	S	Issues program termination, decelerates to a halt
Axis Status	SA1 or nSA2	Returns axis specific status (i.e. limit and home states)
Drive Status	SD1 or nSD2	Returns drive specific status (i.e. type of amp fault)
System Status	SS	Returns general system status and operation

Command	Name	RS-232C Syntax

IS **Tell Input States** **nIS**

Returns the current state (on or off) of the 8 inputs and any of the Optos configured as digital 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 *A1F6<cr> with the input conditions shown in this table.

OPTO Positions (configured as Inputs)								Inputs							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
on	off	on	off	off	off	off	on	on	on	on	on	off	on	on	off
A				1				F				6			

OPTO positions configured as Analog I/O or Outputs are treated as though they are off. (Returns a zero) They are not masked by the control when calculating the hexadecimal values. Your computer program will have to decode the hexadecimal number to determine the state of any each input.

K **Kill** **nK**

Issuing the K command causes the control to abruptly stop commanding further motion and terminates 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 more controlled halt.

MN **Model Number** **MNn,n**

Returns the unit model number over RS-232C. MN command responses are:

*S6961, *S6962, *B8961, *B8962, *961, *962

OS **Tell Output States** **nOS**

Returns the current state (on or off) of the 8 Outputs and any of the Optos that are configured as digital Outputs. The status is returned as a four digit hexadecimal number, preceded by an asterisk.
Example: OS returns *A1F6<cr> with the Output conditions shown in this table.

OPTO Positions (configured as Outputs)								Inputs							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
on	off	on	off	off	off	off	on	on	on	on	on	off	on	on	off
A				1				F				6			

OPTO positions configured as Analog I/O or Inputs are treated as though they are off. (Returns a zero) They are not masked by the control when calculating the hexadecimal values. Your computer program will have to decode the hexadecimal number to determine the state of any each input.

Command	Name	RS-232C Syntax

PA **Tell Absolute Position** **nPAi**

Reports the current position for each axis, in user units, where i is the axis number.
Example: PA1 returns *+10.000 (the position of axis one).

RS **Reset System**

RS 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.

S **Stop** **nS**

Terminates program execution and immediately decelerates each motor to a halt (at a rate set by the SR command). Functions the same as the pressing ESC key on the IDC keypad or activating an input defined as a Stop input.

Command	Name	RS-232C Syntax

SA Tell Axis Status **nSAi**

i=1 for Axis 1 status i=2 for Axis 2 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. See Chapter 3, *Programming Your Application*, for more information on using the Status Commands.

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, not yet implemented
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

Command	Name	RS-232C Syntax

SD **Tell Drive Status** **nSDi**

i=1 for Axis 1 status i=2 for Axis 2 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.

Example: SD1 returns *0010<cr>. This means Axis 1 is enabled, in position mode, and is not faulted. See Chapter 3, *Programming Your Application*, for more information on using the status commands.

Status															
16 off	15 off	14 off	13 off	12 off	11 off	10 off	9 off	8 off	7 off	6 off	5 on	4 off	3 off	2 off	1 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. The B896n requires a reset (RS command, warm boot input, or power cycle) to clear a following error.
I/V Fault *B8961/2 only*	2	1= Over Current or Over Voltage fault (requires reset to clear) 0= At power up and after reset.
Thermal Fault *B8961/2 only*	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= Drive enable signal is active (see also EA1) 0= Drive enable signal is inactive (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, call factory for details) 0= Amplifier in position mode. (factory default)
Amplifier Fault	9	1= The amplifier is faulted. The S696n requires a power cycle to reset the amp. The B896n can be reset via a RS command or a warm boot input. 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

Command	Full Name	RS-232C Syntax

SS **Tell System Status** **nSS**

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. See Chapter 3, *Programming Your Application*, for more information on using the Smart Drive status commands.

Example: SS returns *0001<cr> means there are no amplifier faults, and no programs running. The Indexer is ready to process any buffered RS-232C command.

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	off	off	off	off	on
0				0				0				1			

Description	bit #	Parameter Definition
Ready to buffer RS-232C commands	1	1= Ready to buffer RS-232 commands 0= In a keypad menu, initializing from a power-up or reset, or unchecked errors exist. Any buffered commands sent will be discarded.
EEPROM 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= Command processing paused waiting for a function key. 0= Not waiting at a FK command.
WT active	5	1= Command processing paused waiting for a input condition. 0= Not waiting at a WT command
TD active	6	1= Command processing paused at a time delay. 0= Not waiting at a TD command
Waiting for IV	7	1= Command processing paused, waiting a variable input. 0= Not waiting at a IV command
Buffer full	8	1= RS-232 buffer 75% full Total Capacity: 2k characters 0= RS-232 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.
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

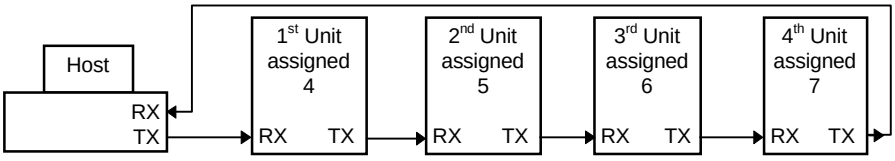
Interface Commands

The following RS-232C commands control program uploading, downloading, deleting, execution, etc. All of these commands are fully defined in this section.

Command	Name	RS-232C Syntax

AA Auto Address AA or AAn

The AA command automatically addresses Indexer 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. AA*n*.

DP Delete Program nDPi

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 the -30k memory upgrade)
Example: DP99 (deletes program number 99).

DR Download Program to RAM nDRi

Begins downloading a program from the host to the control's RAM, rather than non-volatile EEPROM 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 an RS-232C host controller which constantly downloads and executes programs. This increases the usable life of the EEPROM.
Range: 1-199 (1-400 with the -30k memory upgrade)
Example: 1DR50 AC4 DE4 VE30 LP6 DI10.5 GO EN 1EP 1RN50
Downloads program #50 to Unit #1's RAM, then runs program #50

Command	Name	RS-232C Syntax

EP **End Program Definition** **nEP**

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

EX **Ends Upload All or Load All** **nEX**

Singles the end of a upload all (UA) or load all (LA) sequence. EX is sent by the Smart Drive to the host after completing a UA. EX is sent by the host to the Smart Drive to terminate a LA.

LA **Load All** **nLA**

Sent to the Smart Drive 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.

LS **List Programs** **nLS**

Lists number of programs, memory usage, and the current available memory of the Smart Drive. Just like **Edit/List** from the keypad.

PR **Define Program** **nPRi**

Starts a program definition. Just like the DR command, but writes the Smart Drives non-volatile EEPROM 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).

OC **Original Configuration** **OC**

Returns the EEPROM 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.

RN **Run Program** **nRN**

This commands any program, by number only.
The RN command does not support the optional program names.
Example: RN25

Command	Name	RS-232C Syntax

SW **Tell Software Version** **nSW**

The control returns its software revision.
Example: 1SW returns *V1.40 <cr>

UA **Upload All** **nUA**

Uploads all setup parameters and programs from unit n. Smart Drive sends an EX to terminate upload.

UL **Upload Program** **nULi**

Range:i=1-199 (program number)
Uploads program number i to the host. The Smart Drive adds the program brackets. (see below)
Example: 1UL2 Uploads program 2 from unit #1.
Response: { [part A] AC4 VE30 DI10.5 GO }

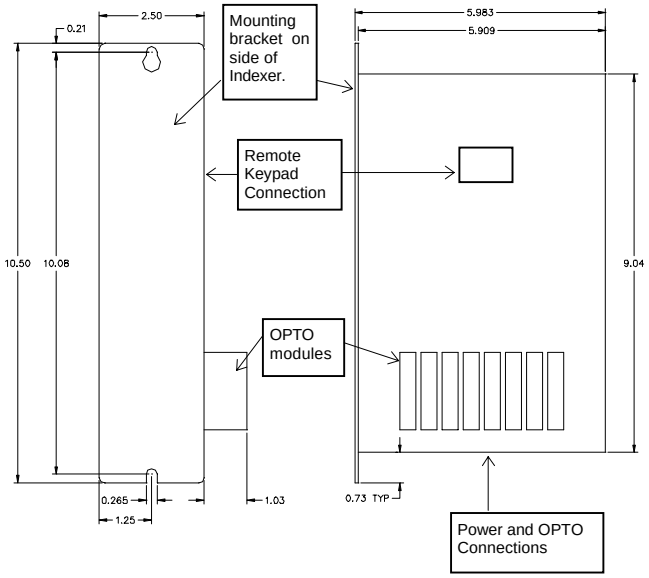
Chapter 6 - Hardware Reference

Mounting Your Indexer

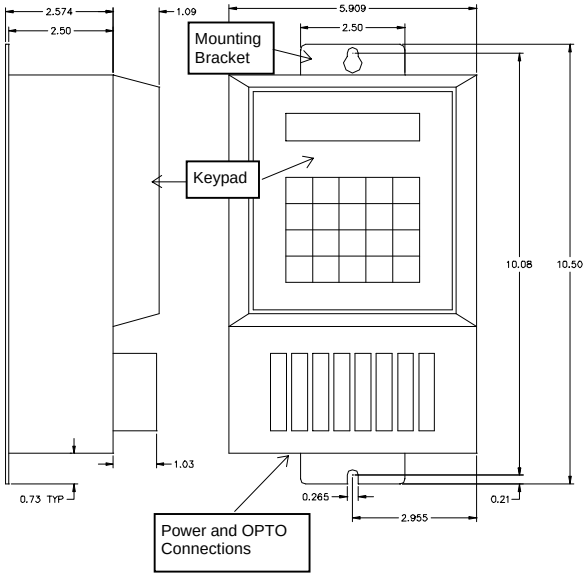
Your 961/2 Indexer arrived with a mounting bracket installed, but the bracket may be moved to either of two positions (shown below) to better facilitate your specific mounting requirements. Moving the bracket to the side allows for minimum width mounting. Installed on the back of the unit, the bracket allows for minimum depth mounting.

To move bracket to either side or back position:
1. Remove 4 screws attaching bracket to unit.
2. Position bracket over four mounting holes at new mounting position.
3. Install same 4 screws you removed in step 1.

Minimum Width Mounting



Minimum Depth Mounting



These general mounting guidelines should be observed:

- When mounting in the minimum width configuration, the vertical clearance between an Indexer and other equipment or surfaces of the enclosure (including other Indexers) should be sufficient to allow for power and OPTO connections. Horizontal clearance should be sufficient to allow for remote key-pad connection and installed OPTO Modules.
- When mounting in the minimum depth configuration, the horizontal clearance should be sufficient to allow for connecting inputs, outputs, drive(s), and encoder(s). In this position, the keypad may be mounted directly on the Indexer using three ball-head mounting screws (included with FP220 Key-pad). Allow enough vertical clearance for connecting power and OPTOs.

Remote Mounting Your FP220 Keypad

The keypad can easily be mounted and sealed to NEMA 4 specifications by using the included mounting gasket and 6-foot communication cable. **Warning: Do not attach the gasket to the keypad.** Attach the gasket with the adhesive side toward the mounting enclosure. A pressure-seal is formed between the gasket and the keypad, while the adhesive maintains the seal between the enclosure and the gasket.

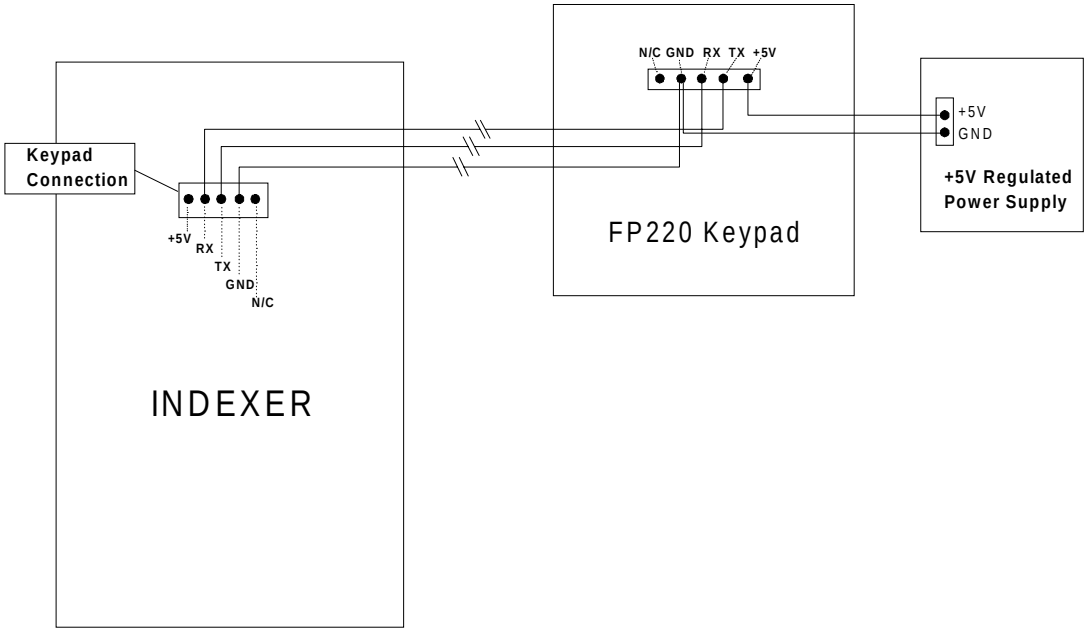
The keypad communicates with the Indexer via RS-232C and this cable may be extended if necessary. For distances greater than the standard remote cable length, a separate +5VDC supply may be required to power the keypad. See drawing below for power supply connections.

An FP220 Keypad Mounting Template is included with every keypad and may also be found in this section.

Please pay particular attention to the **CAUTION** on the template.

Extending the Cable Length to Your Keypad

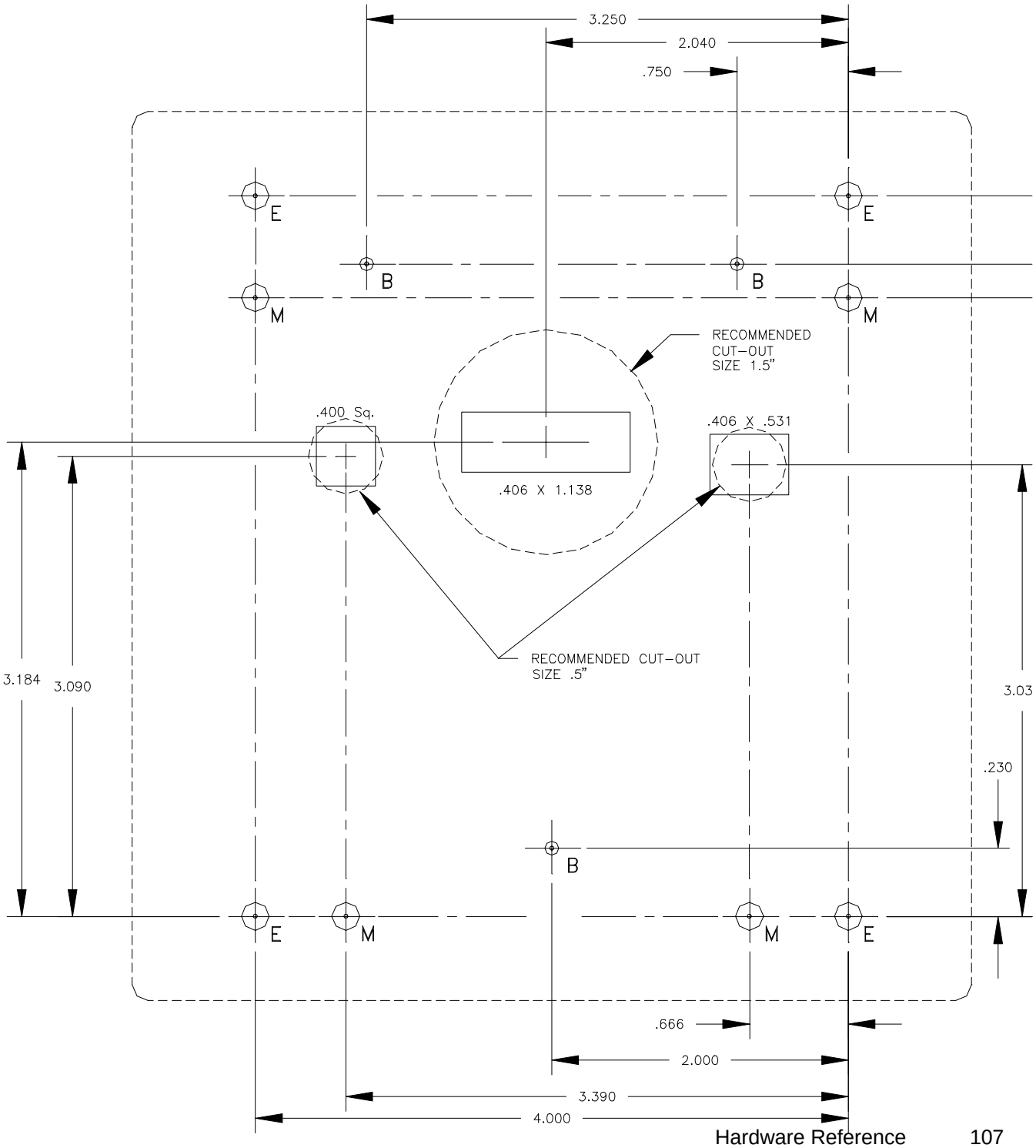
The keypad requires a stable +5VDC to operate its electronics. Since voltage drop may occur at long cable lengths, it may be necessary to connect an independent power supply as shown below.



FP220 KEYPAD MOUNTING TEMPLATE

CAUTION: Your FP220 Keypad will be DAMAGED if mounting screws extend more than 0.2" into the keypad.

E	3/16" CLEARANCE HOLES (4)	CUT THESE HOLES FOR 6/32 CAPTIVE MOUNTING.
M	3/16" CLEARANCE HOLES (4)	CUT THESE HOLES FOR M3.5 CAPTIVE MOUNTING.
B	5/32" CLEARANCE HOLES (3)	CUT THESE HOLES FOR BALL-HEAD REMOVABLE MOUNTING.



961/962 Hardware Specifications

Input Power Requirements

120 VAC single phase, 50/60 Hz., 2.0 Amps maximum.

Output Power Available

12 VDC internal power supply, 250 mA maximum output current.
5 VDC at 200 mA available for encoder.

Environmental Requirements

Ambient Temperature: 0 - 50° C

Humidity: 0% to 90% non-condensing.

Drive Signals

Step, Direction, & Shutdown Outputs: Optically isolated. Low signal <0.8 VDC, high signal >3.5 VDC, ±60 mA. Active high. Step pulse width is 0.8 to 10 μsec (depending on drive resolution setting).

Drive Fault Input: Optically isolated, TTL level, internal 1.0 Ω pull-up to +5 VDC.

Position Range: ±0 to 2,147,483,647 steps. Absolute and incremental.

Velocity Range: 1 to 1,250,000 steps/sec.

Acceleration Range: 1 to 20,000,000 steps/sec^2

OPTO-compatible I/O

8 positions support OPTO-22 (G4) digital, and Grayhill (G5) analog and temperature modules (see ordering information)

Encoder Interface

Optically isolated, differential 5 VDC, 2 MHz max. (post-quadrature).

Outputs

8 programmable outputs. Open collector, sink current - 100 mA max.

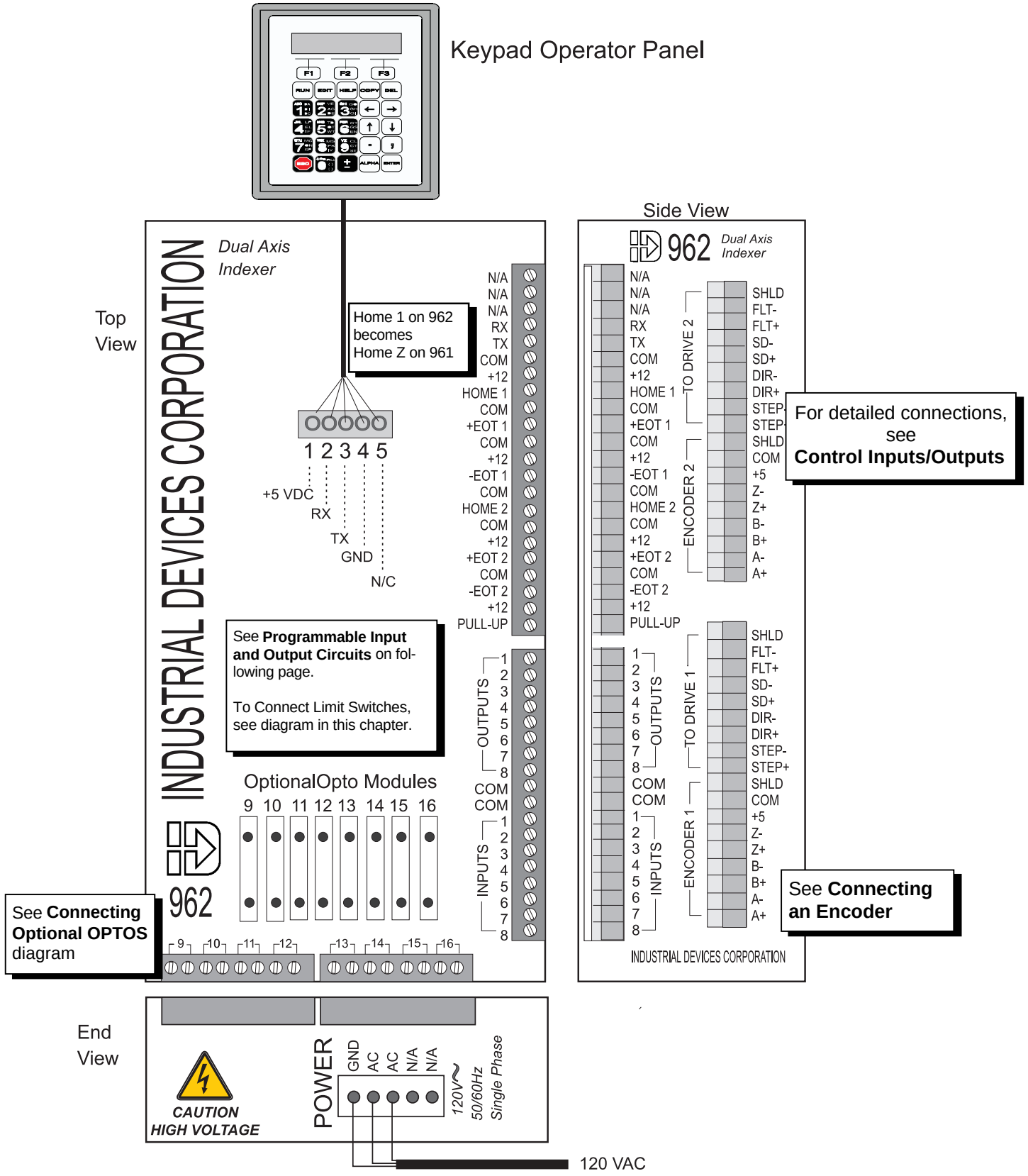
Inputs

8 programmable inputs + 2 Limits + 2 Home. 24 VDC max. Optically isolated, can be pulled up to internally isolated 12 VDC supply. 12 mA current required.

Programming

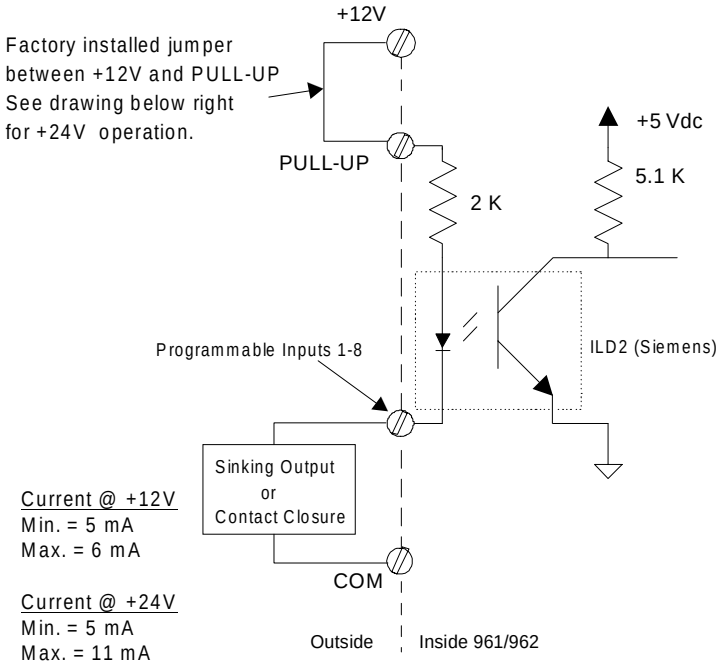
IDeal™ programming language. Program from the keypad operator panel, or via your PC using our Windows-compatible IDCMotion™ Application Developer software (included).

961 and 962 Hardware Connections

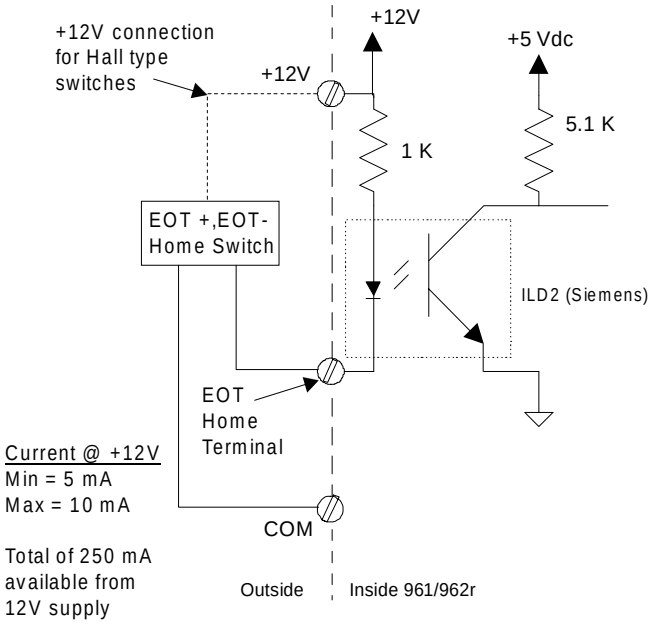


961 and 962 Input and Output Schematics

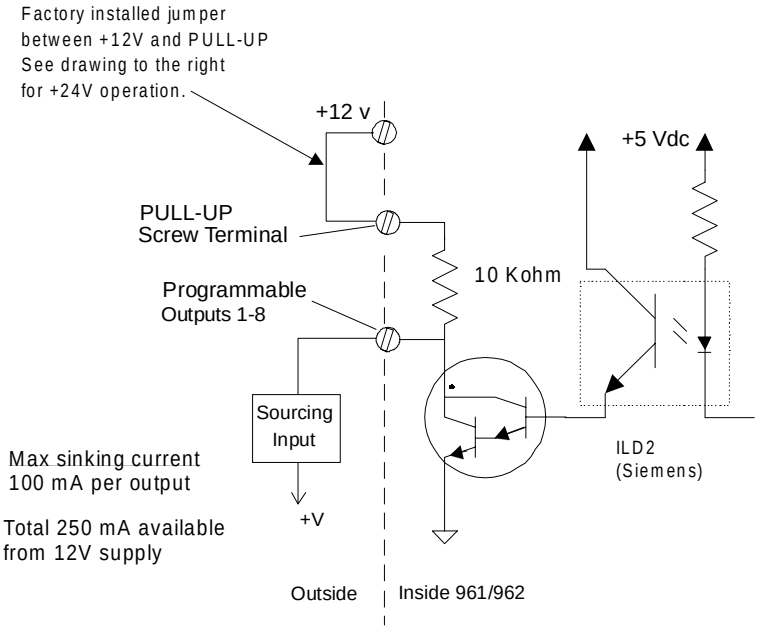
Programmable Input Schematic



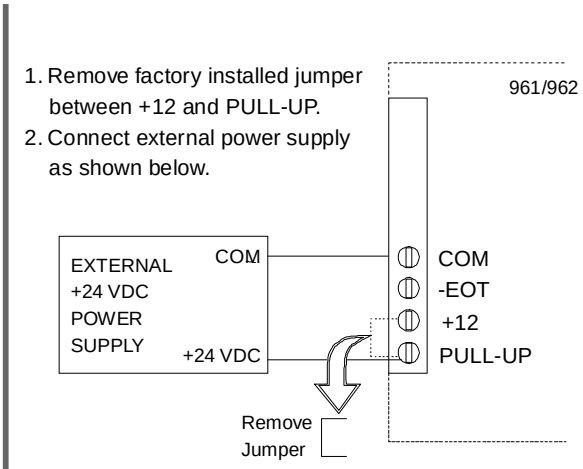
EOT and Home Switch Schematic



Programmable Output Schematic



Connecting a +24V Power Supply



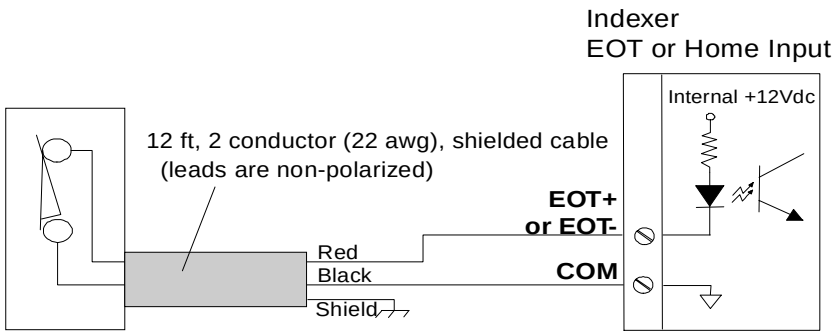
Connecting Limit Switches to the Indexer

(RPS and RP Switches Supplied by IDC)

RPS-1 is a normally open switch, used as a Home only.

RPS-2 is normally closed.

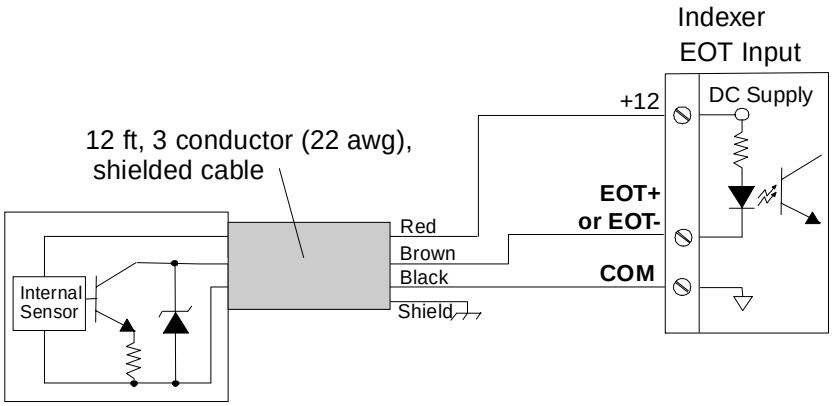
RPS switches are mechanical Reed switches, which are less expensive, have less wiring, and are potted.



RP-1 is a normally open switch, used as a Home only.

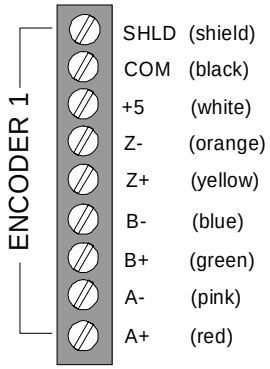
RP-2 is normally closed.

RP switches are electronic Hall Effect switches, which have longer life, and use LEDs.

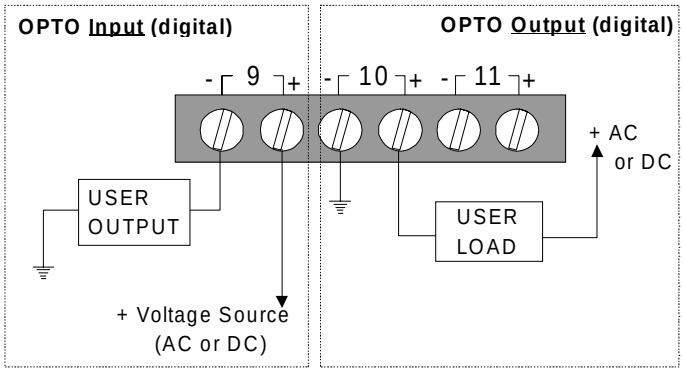


Connecting an ENCODER to an INDEXER

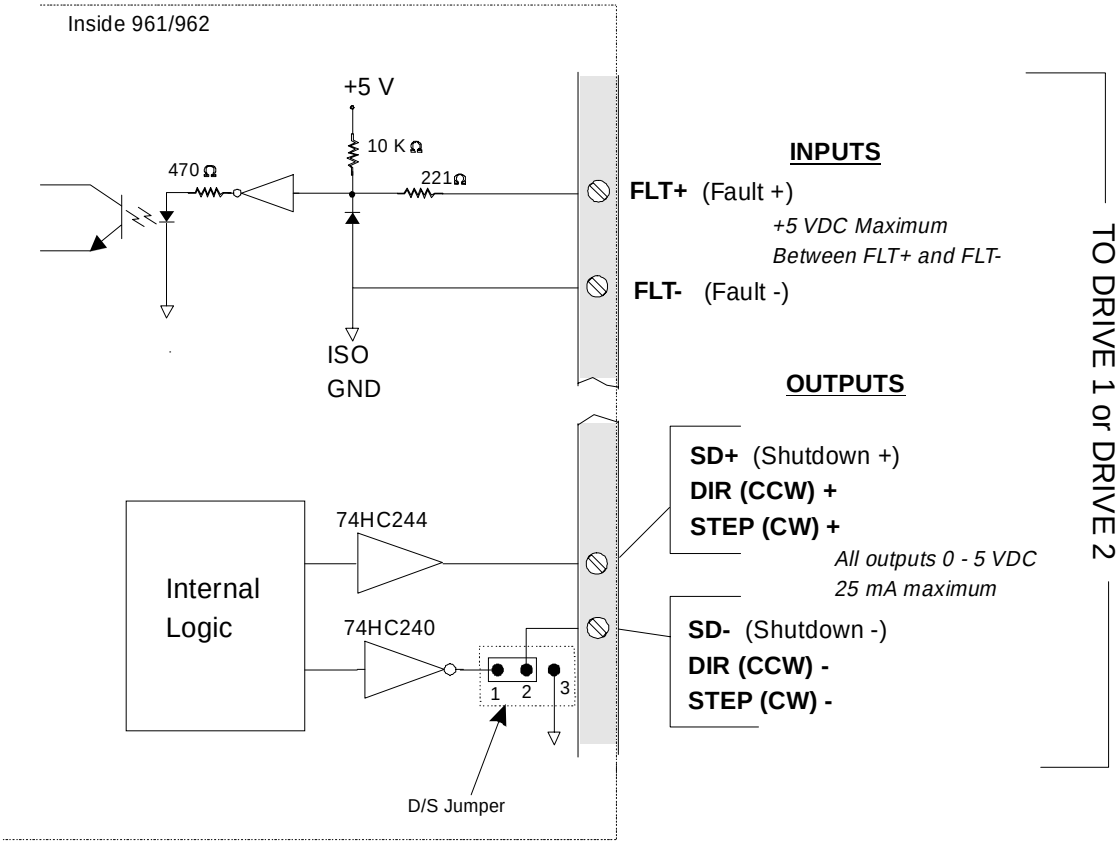
ENCODER 1 and ENCODER 2 are connected identically (only the 962 has both ENCODER 1 and ENCODER 2). Color codes are for IDC encoders only. Use signal names for other manufacturer's encoders.



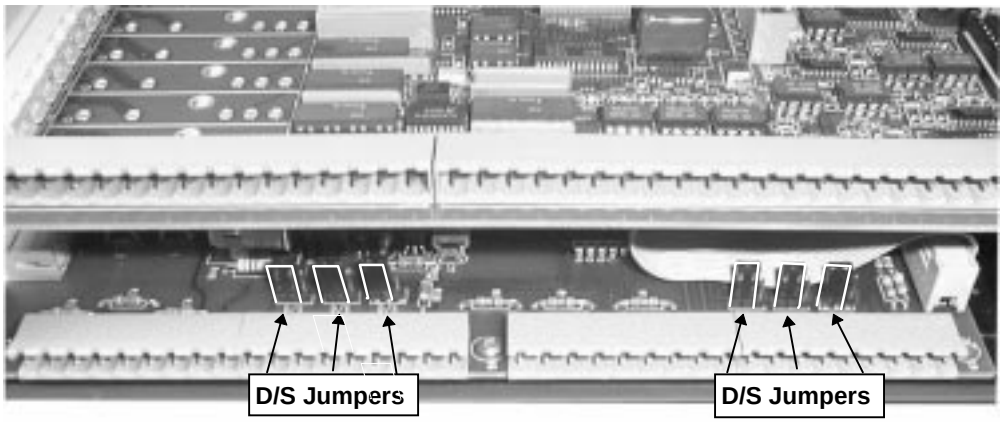
Connecting Optional Opto Modules



Control Inputs/Outputs



D/S (Differential/Single-Ended) Jumpers

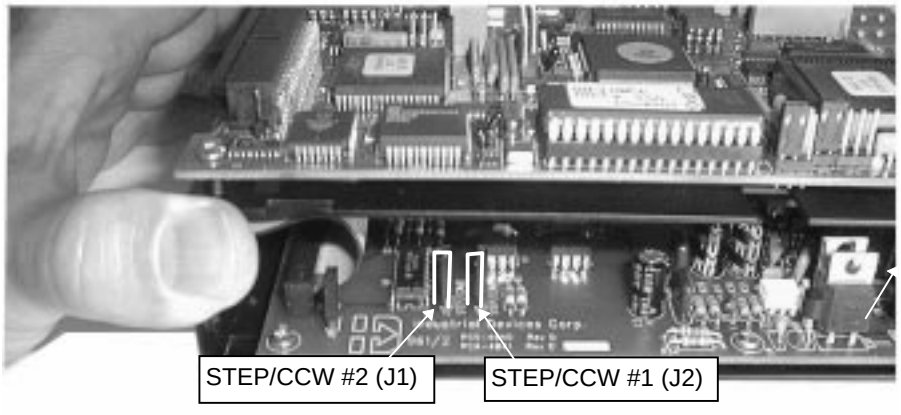


Changing (Differential/Single-Ended) Jumper Settings

D/S jumpers are shown in the D position, which is the way they are shipped from the factory. Jumpers are located on the lower board just behind the Encoder and Drive connections.
To change jumper to the S (Single-Ended) position:

1. Remove Indexer cover by removing 4 cover screws, 2 on each end of the unit.
2. Carefully lift cover straight up and away from the Indexer.
3. Using needle-nosed pliers, pull jumper straight up, move it to the right and reinsert it on pins 2 and 3 (the S position).
4. Repeat step 3 for all D/S jumpers. There are 3 jumpers on a single axis Indexer (961), and 6 jumpers on a dual axis Indexer (962). All jumpers must be changed at the this

STEP/CCW Jumpers

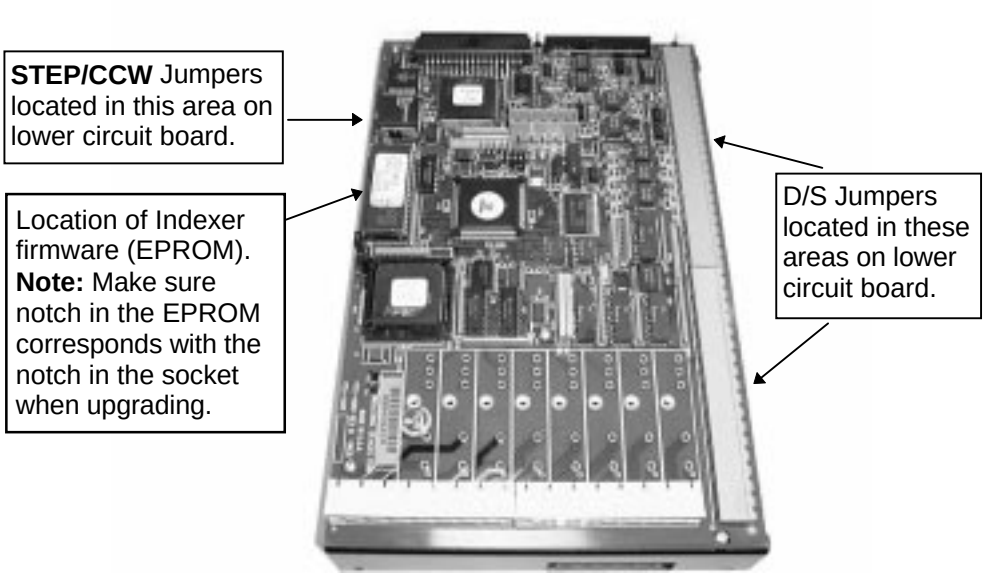


Changing STEP/CCW Jumper Settings

STEP/CCW Jumpers are shown in the STEP position, which is the way they are shipped from the factory. Jumpers are located on the lower board on the side opposite the D/S jumpers. General location shown in photos above and below.
To change jumpers to the CCW positions:

1. Follow steps 1 and 2 on previous page for Indexer cover removal.
2. Using needle-nosed pliers, pull jumper J1 straight up, move it to the CCW position and push it straight down.
3. Repeat step 2 for jumper J2.

Indexer Shown with Cover Removed



Troubleshooting

The following table will help you isolate some of the more common application problems when using Indexers:

Symptoms	Probable Causes	Possible Remedies
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. Also check the drive resolution setting in the EDIT > SETUP > MOTOR > D-RES menu.
Motor stalls at high speeds	Commanded velocity is too high for system capability.	May have to reduce velocity.
Motor stalls during acceleration	Motor current is incorrect (Stepper Drive only). Acceleration rate is too high for the system capability.	Check motor current setting (Stepper). Reduce your acceleration, or use a motor with higher torque.
Motor moves in the wrong direction.	The system's sense of direction is reversed.	Change the controls direction from the EDIT > SETUP > MOTOR > DIR 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.
Controller does not recognize OPTO outputs.	OPTO positions default to inputs.	Configure as output from the EDIT > SETUP > I/O > OPTOS menu.

IDC Product Support

Factory Authorized Distributors

IDC has more than 60 factory trained and 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 Industrial Devices' systems, as well as other complementary equipment. Contact us at (800) 747-0064 or (415) 382-4300 for the name of the distributor in your area.

Regional Offices

IDC Distributors are supported by local, direct IDC Regional Managers. There are currently 8 IDC regional offices in North America. IDC Regional Managers are available to assist with unusually demanding application, present on site customer seminars, determine custom product needs, or respond to high volume requirements.

Toll Free Technical Support

Industrial Devices 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 Applications Engineering department is open from 6am to 5pm Pacific Time, Monday through Friday. The toll free number is (800) 747-0064. Outside of the United States call (415) 382-4300. The technical support fax number is (415) 883-2094. 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. Low volume requests are complimentary. Check the website for CAD drawings that may be downloaded.

Website

Warranty & Repairs

Industrial Devices Corporation (IDC) 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 IDC, are not subject to the terms of this warranty.

IDC maintains a repair facility at its factory in Novato, California for products manufactured by IDC. Prior approval by IDC 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 IDC (1-800-747-0064) for an RMA#.
4. Provide information describing the nature of the failure. The better the information, the faster we'll have your problem resolved.
5. Ship unit prepaid to:

   Industrial Devices Corporation
   64 Digital Drive
   Novato, CA 94949

Appendix A: IDC Actuator Ratios

Configuring *inch* & *mm* Units on Indexers Used With IDC Actuators

One of the first steps in setting up an IDC Indexer 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 motors revs per 1 distance unit. Several other examples are available in the *Configuring Your System* chapter of this 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…).

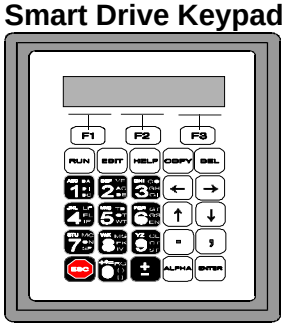
Shown below are the three different ways to change the RATIO in an Indexer. Please note that some ratios cannot be entered via the keypad or Application Developer. The keypad and Application Developer screens only support up to 5 digits in each RATIO number. Via RS-232C, up to 8 digits can be entered for each number. The rounding error caused by only being able to enter 5 digits is very minimal for most actuators and stroke lengths, and is often much less than the positional uncertainty caused by mechanical backlash and windup.

Methods For Configuring Ratio

There are three methods for entering the ratio information. The keypad is the quickest method, if your system includes that option.

1. Indexer Keypad [EDIT-SETUP-MECH-RATIO Menu]
2. IDC Motion™ (Application Developer) [*Setup, Axis* Menu]
3. Direct RS-232C connection [GR Command]

Smart Drive Keypad



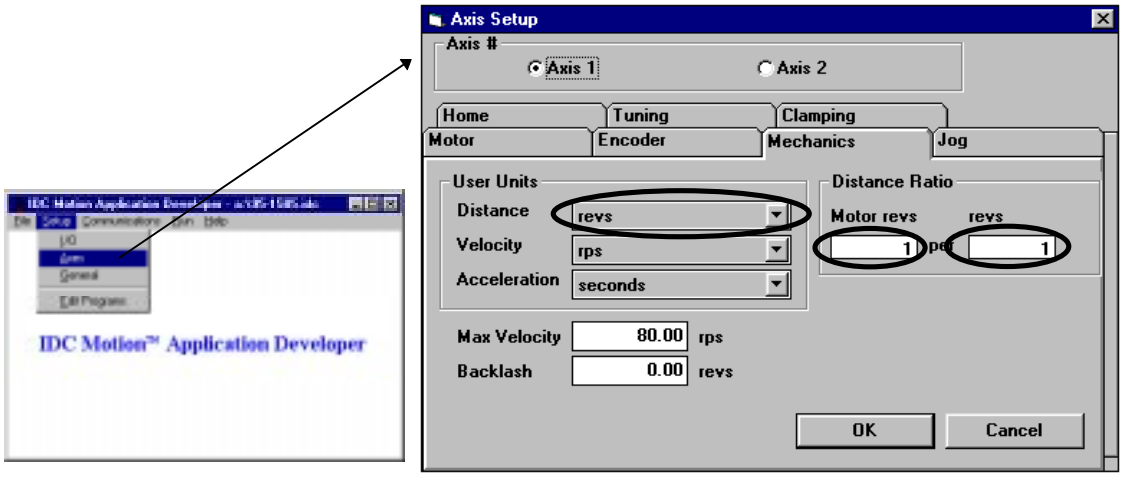
RATIO

EDIT > **SETUP** > **MECH** > **RATIO**
Default: 1 to 1

- - - Axis One Ratio - -
←↑ 1 to 1 ↓→

These two integer values set the number of ***motor rev's*** per ***distance unit*** (i.e. inch, mm, cm, etc.)

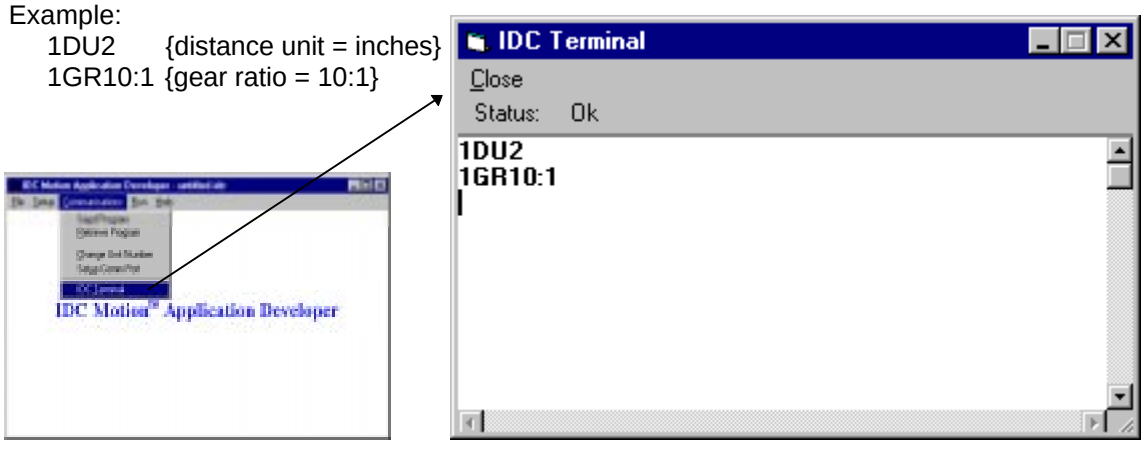
Application Developer



Please note that your entire application (setup parameters and programs) must be downloaded to the Indexer before the new RATIO scaling will be used. You cannot download only a new RATIO from Application Developer.

RS-232C Terminal

Example:
1DU2 {distance unit = inches}
1GR10:1 {gear ratio = 10:1}



Gear Ratio 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 entering units of *Inches* and *mm* can be found in the following actuator ration tables.

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 1524
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
R3-502	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	15 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)

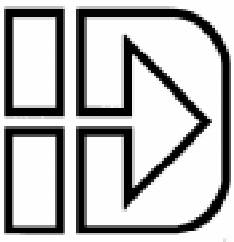
Instruction	Example
1) **Select User Units** Select a preferred unit-of-measure for linear distance. This will be used for programming distance, and can be used for velocity and acceleration as well.	*centimeters*
2) **Determine Overall Mechanical Ratio** Look up actuator mechanical "inch" ratio. Units 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 the Smart Drive - it might be necessary to round of the number from 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], [SETUP], [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 Smart Drive** The Smart Drive is now ready to program in your own User Units. Distances will match the units configured above.	*DI10.0 GO* *(moves 10.0 cm)*

Index

T

V

W



INDUSTRIAL DEVICES CORPORATION
64 Digital Drive • Novato, CA USA 94949-5704
(800) 747-0064 • FAX (415) 883-2094
OUTSIDE THE U.S. CALL (415) 382-4300

E-mail: INFO@IDCMOTION.com
World Wide Web: http://www.idcmotion.com