KOLLMORGEN®

Because Motion Matters™

This is a Discontinued Product

Contact Kollmorgen Customer Support at 1-540-633-3545 or email us at support.kollmorgen.com if assistance is required.

IDC

SmartStep™

Microstepping
SmartDrive

User's Manual

P/N PCW-5008 Ver.2 07/03

Revision History

Version 1.0 - May 2003

IDC strives to maintain effective communication with all users and potential users of our products. If you have any questions or concerns regarding this technical manual or the product it covers, please contact:

Danaher Motion
7C Raymond Avenue
Salem, NH 03079

TEL: **(800) 277-1066** FAX: **(603) 893-8280**

OUTSIDE THE U.S. CALL **(603) 893-0588**

WEB SITE: **www.idcmotion.com**

EMAIL: sales@idcmotion.com

Table of Contents

Chapter 1 - Overview

IDC's SmartStep Microstepping Drive is the first programmable, all digital motion controller to offer so many features in such a small package.

The SmartStep provides unmatched flexibility when teamed up with the optional keypad operator interface. We recommend at least one 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 SmartStep can also be programmed over RS-232C with IDC's user-friendly *Application Developer* software. Serial communication commands may be found in Chapter 8.

This manual has been designed to help you successfully install, program, and operate your SmartStep. If you have any questions that are not adequately answered in this manual, please call our factory application engineers at 1-800-544-8466.

Use Chapter 2 - Shipping Contents to help verify that you have received everything you ordered.

Quick Start (Chapter 3) will help you quickly confirm basic system operation.

Chapter 4 - *Using the Keypad* describes IDC's keypad operator interface.

Chapter 5 - *Configuring Your System* covers software and hardware configuration of your application. It includes step-by-step keypad instructions on entering setup parameters. This chapter covers initial motor settings, I/O configuration, mechanical setup, and other special parameters of your system. IDC's *Application Developer* also follows the same menu structure described in Chapter 5.

In Chapter 6 - *Programming Commands* provides an alphabetical list of SmartDrive commands including syntax, ranges, defaults, and programming examples for each command.

Chapter 7 - *Programming Your Application* provides detailed program and application examples and strategies. Other topics include variable usage, user menus, math functions, and analog I/O. Our *IDeal*™ command language is generally regarded as the easiest motion control language in the industry. It is both easy to remember and intuitive, without sacrificing flexibility or power.

Chapter 8 - *Programming with Serial Communication* is for users who plan to configure and program the SmartStep in an RS-232C or RS-485 hosted mode. IDC's *Application Developer* program follows a standard Windows dialog-box structure for straightforward cofiguration and programming of the SmartStep. This section also covers RS232C command syntax and definitions for users who are not using Windows.

Chapter 9 - *Hardware Reference* provides SmartStep mounting information, specifications, detailed I/O schematics, and IDC motor data.

Chapter 10 - *Troubleshooting* lists some common application problems along with their symptoms and solutions.

Included with this manual is the *IDC Application Developer CD*. IDC's *Application Developer* is automatically installed on your hard drive by running the setup program on the *IDC Application Developer CD*. This disk also includes a readme file containing the latest information on software features. The readme file also contains a listing of demo program files included with *Application Developer*.

Chapter 2 - Shipping Contents

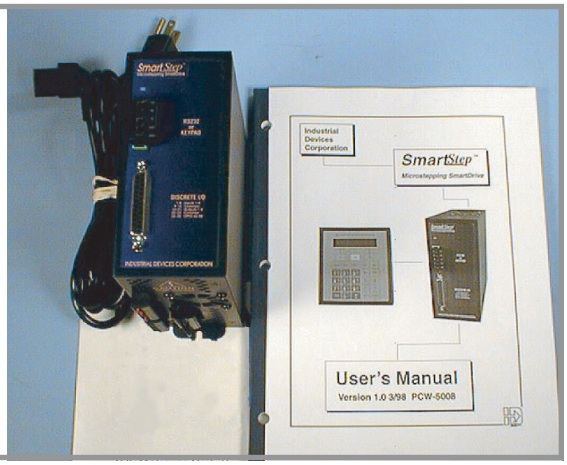
Your SmartStep will arrive equipped as listed below. If any parts or accessories are missing, please call IDC Customer Support at: (800) 227-1066.

If you ordered a SmartStep with Keypad, you will receive:

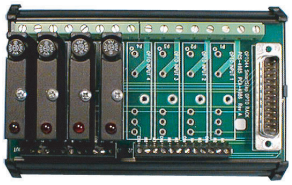
1. SmartStep, SmartStep23, or SmartStep240 (-MD will be added if you ordered a minimum depth mounting, e.g. SmartStep-MD)
2. Keypad
3. AC power cord for SmartStep
4 Remote cable for Keypad
5. Keypad mounting gasket
6. Keypad mounting template
7. CD with Application Developer software and product manual

If you ordered a SmartStep only, you will receive:

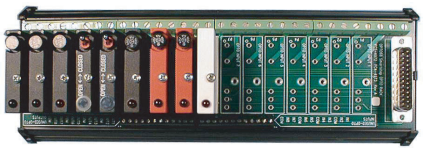
1. SmartStep, SmartStep23, or SmartStep240 (-MD will be added if you ordered a minimum depth mounting, e.g. SmartStep-MD)
2. AC power cord for SmartStep
3. CD with Application Developer software and product manual



SmartStep Accessories Currently Available



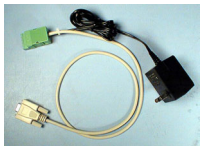
OPTO44 Rack
(May be ordered with or without OPTO modules installed)



OPTO88 Rack
(May be ordered with or without OPTO modules installed)

DB25BO Screw Terminal Breakout Board

SS-PNP-BO Screw Terminal Breakout Board



PCS-5004
PC-Keypad Cable for copying programs between Keypad and PC.

SS-IO (2ft. cable)
SS-IO-6 (6ft. cable)

SS-RS232
SmartStep to PC Cable

Chapter 3 - Quick Start

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

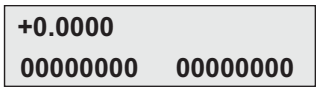
The Keypad (FP220) is highly recommended for the easiest possible setup and programming of your application. For **Quick Start** it is assumed that a keypad will be used, even though serial communication may be used later.

Quick Start

1. Connect your keypad and motor to the SmartStep. If you have questions about connecting your motor, refer to Chapter 9, *Hardware Reference*.

2. Connect the power cord to the AC connection on top of the SmartStep. Power is applied to the SmartStep when the power cord is plugged into the AC power source.

3. Apply power. The "ON" LED on front of the SmartStep will be green. The keypad LCD display briefly shows Model # and Firmware Revision, then changes to the main diagnostic display as shown to the right.

   Note: The LCD display may require adjustment for better viewing. If so, please refer to "Adjusting Contrast" in Chapter 4 - *Using the Keypad*.

4. On the keypad, press **EDIT > SETUP (F2) > MOTOR (F1) > TYPE (F1) > STEPER (F1)**. The display should read as shown to the right.

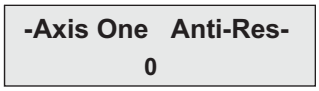
   For steps 5 through 12, use Motor Reference - 1 on the following page for current, unloaded anti-res, and inductance settings.

5. Select **CURRENT (F1)** (SmartStep is shipped from the factory with motor current set at zero).

6. Enter the appropriate current setting for your motor. Press **ESC**.

7. Select **A-RES (F2)**.

8. Enter the **Unloaded Anti-Res** setting for your motor. Press **ESC**.

9. Select **INDUCT (F3)**.

10. Select **HIGH** or **LOW** inductance for your motor. Press **ESC** until you return to the main display.

Your motor should now have torque. If your mechanical system allows it, check for

torque by trying to turn the motor shaft by hand.

11. Press **RUN > JOG (F2)**.

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

12. Jog the motor by selecting **LO** or **HIGH.** Confirm that the motor turns in both directions by pressing ← and →.

13. If your motor is moving when you press the arrow keys, your system is set up correctly. If your motor does not move, see Chapter 10, *SmartStep Troubleshooting*.

14. For complete setup instructions, see Chapter 5, *Configuring Your System*, or use the Setup Wizard in the *Application Developer* software.

Motor Reference - 1

MOTOR **SERIES (T)**	CURRENT @120	CURRENT @240	Inductance Setting	Unloaded Anti-Res	MOTOR **PARALLEL(V)**	CURRENT @120	CURRENT @240	Inductance Setting	Unloaded Anti-Res
S12	1.0	1.0	LOW	29	S12	2.0		LOW	29
S21	1.2	1.2	HIGH	30	S21	2.3		LOW	27
S22	1.5	1.5	HIGH	28	S22	3.0		LOW	24
S23	1.7	1.7	HIGH	25	S23	3.4		LOW	22
S32	2.8	2.8	HIGH	22	S32	5.6		LOW	18
S33	3.5	3.5	HIGH	21	S33	7.0		LOW	17
S42	6.0	4.0	LOW	16	S42	7.9		LOW	12
P21		0.7	HIGH	30	P21	1.3		LOW	27
P22		1.0	HIGH	28	P22	2.0		LOW	24
P31		1.5	HIGH	27	P31	2.9		HIGH	23
P32		1.6	HIGH	24	P32	3.3		HIGH	20
P33		2.0	HIGH	22	P33	4.0		HIGH	18
P41		2.8	HIGH	21	P41	5.7		HIGH	17
P42		3.3	HIGH	18	P42	6.6		HIGH	17
P43		3.3	HIGH	15	P43	6.6		HIGH	14
K31		1.5	HIGH	24	K31	2.9		HIGH	20
K32		1.7	HIGH	22	K32	3.4		HIGH	18
K33		1.7	HIGH	20	K33	3.3		HIGH	16
K41		2.8	HIGH	18	K41	5.7		HIGH	14
K42		3.2	HIGH	17	K42	6.4		HIGH	13
K43		3.3	HIGH	15	K43	6.6		HIGH	11

Chapter 4 - Using the Keypad

This chapter will help the first-time user understand the basics of using the Keypad. The keypad was designed to provide operators the quickest possible way to configure an application, and though it functions primarily as an operator interface, it is equally effective as a programming and troubleshooting tool. If you have chosen to program your application using RS-232C, you may want to skip this chapter.

Keypad Hardware Features describes physical settings and adjustments that facilitate operator access for programming and best viewing of the keypad display. *Keypad Features* and *Description of Keys* provide an overview of functionality.The *Keypad Menu Structure* section gives the programmer a broad overview of how the setup and programming menus operate. Detailed information about each setup parameter is presented in Chapter 5 - *Configuring Your System*.

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

Here are some of the operating functions you can program with the SmartDrive:
- 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 SmartDrive s operator interface, see Chapter 6 - *Programming Commands*, and Chapter 7 - *Programming Your Application*.

Keypad Features

- Easy-to-read, two-line, 40-character, back-lit display
- Can be sealed to IP65 (NEMA 4) washdown environment
- Large, scratch-proof keys with audible and tactile feedback
- Connects to SmartDrive and other controls with remote cable
- Allows application programs to be copied from one SmartDrive to another, or to/from a PC.

Keypad Hardware Features

Setting DIP Switches to Limit Access to Keypad Menus

Four DIP switches on the back of the keypad provide 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 firmware or setup parameter in the SmartDrive)

DIP Switch Settings				Keypad Functionality
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.				

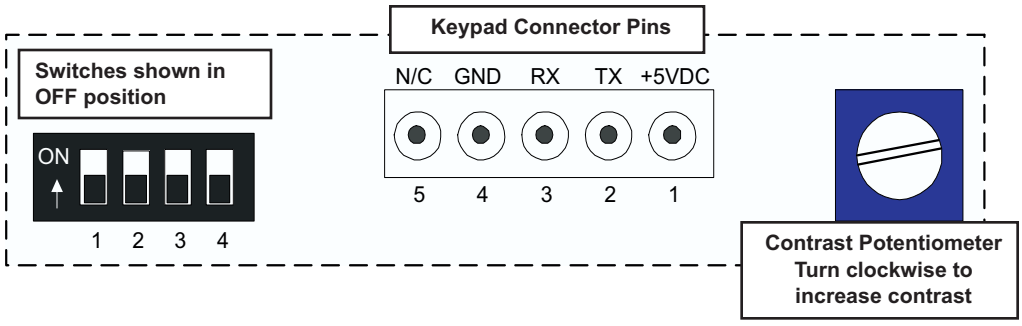
Notes: Power must be cycled before DIP switch setting changes take effect.
Access to the **JOG** menu can be enabled or disabled from software.

Using Passwords to Limit Access to Keypad Functions

Another method of limiting access to keypad functions is to assign passwords when configuring your setup parameters. See Configuring Your Miscellaneous Setup Parameters in Chapter 5 for more information on passwords. Please note that DIP switch settings have priority over passwords.

Adjusting Contrast

On the back of the keypad there is a plastic potentiometer, adjustable with a flathead screwdriver. This is used to adjust the contrast on the LCD display. If the SmartDrive 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.



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. Refer to page 9-3, Hardware Reference, for information on mounting the keypad and extending cable length.

Functions of the Keypad Keys

F Keys (F1-F2-F3)
Used as Menu selectors. Used with numeric keys to select commands in the program editor.

Programmable as operator menu selections. See the FK command for information on using the function keys within a program.

0-9 Keys
Enters numbers. Used with ALPHA to select characters, and with F Keys to select commands in the program editor.

Menu Keys

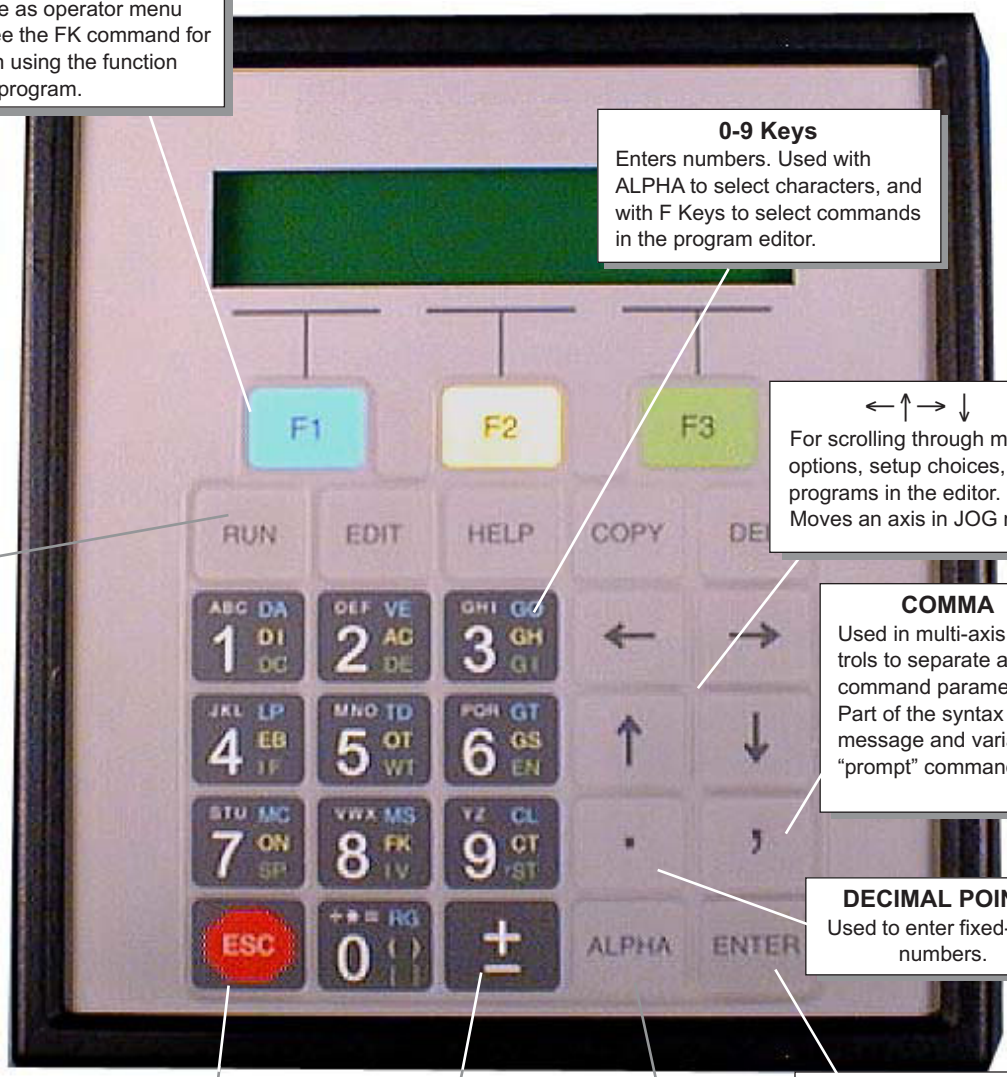
RUN - Runs a program, jogs an axis, or accesses Test/Debug functions.

EDIT - Edits Setup parameters and programs, lists programs, & resets position counter.

HELP - Provides help on keys, menus, and command syntax.

COPY - Copy programs between keypads, between keypad and PC, or from program to program within a SmartDrive.

DEL - Deletes characters in the editor or deletes entire programs from memory.



←↑→↓
For scrolling through menu options, setup choices, and programs in the editor. Moves an axis in JOG mode.

COMMA
Used in multi-axis controls to separate axis command parameters. Part of the syntax in message and variable "prompt" commands.

DECIMAL POINT
Used to enter fixed-point numbers.

ENTER
Saves parameters that have been typed into a configuration or the program editor. Enters a space in the program editor.

±
Selects the direction of motion in programs, and may be used in math programs.

ALPHA
ALPHA plus a numeric key selects alphabet characters and other special characters (see page 4-13)
Pressing:
ALPHA +1 enters "A"
ALPHA +1+1+1+1 enters "a"
ALPHA + ↑ or ↓ enters ? ! @ # % & < > : ; \ ' " _ | ←↑→↓

ESC
Stops a program or moves back one menu level. Exits and saves a program in the program editor.

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.

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

Menu Options	Main Menu Keys and Associated Menus - Press Once to Begin				
	RUN	EDIT	HELP	COPY	DEL
	PROG (F1) Run programs by name or number.	**PROG (F1)** Edit or write programs.	**In Main Menu:** Provides help on the functions of RUN, EDIT, and COPY. Gives SmartStep, keypad, firmware, and FPGA versions.	**PROGRAM (F1)** To copy programs within a control.	**PROGRAM** Deletes an entire program, or characters in the editor.
	JOG (F2) Jog either axis at low or high speeds - Press F1 or F2, and any arrow key (←↑→↓)	**SETUP (F2)** Configure system components and operating limits.	**In Menus:** Provides help on moving about in menus.	**TO PAD (F2)** To upload a control's memory to the keypad.	
	TEST (F3) Run programs in trace mode, amplifier shutdown and reset, test outputs, or moves.	**POS (F3)** Reset axis position to zero? YES (F1) NO (F3)	**In Sub-Menus:** Explains setup choices.	**FROM (F3)** To download keypad memory to a control.	
		LIST (↓) (F1) Directory of stored programs, memory usage and available space.	**In the Editor:** Provides command descriptions.		

If a menu has more than three options, arrows will appear on both sides of the display to indicate that more options are available. Press the appropriate arrow key to cycle, one display at a time, through all options in that menu. To exit 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.

Using the RUN Menus

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

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

Select PROG To Run a Program

To run an existing program by program number

```
↑ RUN PROGRAM ↓
> 5
```

1. Press F1 (PROG).
2. Press program number 1-199 using numeric keys (1-400 with 30K 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 the motor

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

1. Press RUN.
2. Press JOG (F2).
3. Press ←↑ and ↓ → keys to jog the motor.

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: .012
```

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 steps 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.

Using 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.

```
↑ TRACE PROGRAM ↓
>_
```

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

```
PR:5  LP:1  GS:0
DI8000
```

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

```
Test Output #1
←↑      ON      ↓→
```

1. Press RUN > TEST > OUTPUT.
2. Press ← → keys to scroll through outputs 1-8.
3. Press ↑ ↓ keys to turn the output ON and OFF.
4. Press ESC and you will be prompted as shown here:
5. Make your selection and the display will immediately return to the Main Display.

```
Save Output States
YES          NO
```

```
+0.0000
00000000  00000000
```

Please use caution when connected to live devices.

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.

```
------ TEST MOVE ------
Axis 1
```

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

SHUTDN (Shutdown)

Selecting SHUTDN allows you to enable or disable the axis. When a drive is disabled, the amplifier is off and your motor has no power. The shaft can easily be manually rotated. RESET is applicable only to the B8961/2.

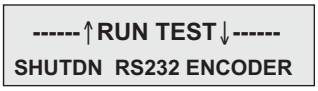
1. Press RUN > TEST.
2. Press ↑ or ↓ until SHUTDN appears above F1.
3. Press SHUTDN.
4. Press ENABLE or DISABLE.

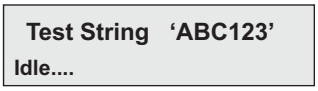
RS232

The RS232 feature allows for testing and debugging of daisy chain terminal communications through the keypad, thus eliminating the need for a PC terminal connection.

Testing Serial Transmit

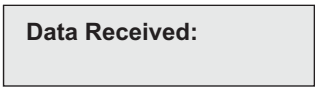
1. Press RUN > TEST.
2. Press ↑ or ↓ until RS232 appears above F2.
3. Press RS232.
4. In the Test Connection menu press TRANSMIT. The SmartDrive will now transmit the string ABC123 every 5 seconds.





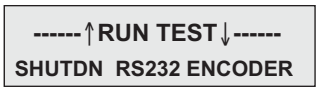
Testing Serial Receive

1. In the Test Connection menu press RECEIVE.
2. Any character received on the terminal port will be displayed on the keypad.



ENCODER

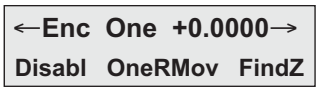
The ENCODER sub-menu allows you to perform three different tests to determine if encoders are working properly. Refer to page 5-13 if your encoder has not yet been configured.



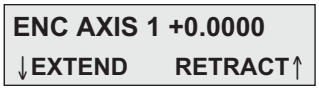
1. Press RUN > TEST.
2. Press ↑ or ↓ until ENCODER appears above F3.
3. Press ENCODER.
4. Press desired encoder test (described below).
5. Use the ← or → arrows to select the axis of the encoder to be tested.

Note: the encoder tests are performed by toggling the F1, F2, and F3 buttons.

Disabl/Enable (F1) 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.



OneRMov (F2) 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.



FindZ (F2) commands the motor to rotate until it finds the Z pulse position. This allows you to accurately center the position of the Home switch.

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 and TUNING:

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

Access the PROG, SETUP, POS, LIST, and TUNING menus by pressing the appropriate function key. Descriptions of these sub-menus follow:

Using the PROG Sub-menu to Create and Edit 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 (1-400 with 30k memory option), but do not use a number that is already being used for another program. If the SmartDrive contains several programs, scroll the list 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 later 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 SmartDrive 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 Number Keys

The programming commands on the number keys have been color-coded to match the color-coded function keys (F1, F2, F3). For example: the yellow function key (F2) will be used to enter all yellow-colored commands on the number keys, e.g. AC command on the number 2 key.

To enter a command:

1. Press the function key that matches the color of the command you wish to enter.
2. Press the number key that contains the desired command.
3. Press the ENTER key to insert a space before entering the next command.

Note: All commands that may be used by the SmartDrive are not on keypad keys. See *Chapter 6 - Programming Commands*, for a listing of all available commands.

Step-by-Step Example of Entering a New Program

(You must be inside the program editor as accomplished in steps 1 through 5 on the previous page)

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

1. Press F2.
2. Press the #2 key. This will enter the AC command.
3. Press the decimal or period key (adjacent to the #9 key).
4. Press the #3 key.
5. Press ENTER.
6. Press F1.
7. Press the #2 key. This will enter the VE command.
8. Press the #2 key.
9. Press ENTER.
10. Press F2.
11. Press the #1 key. This will enter the DI command.
12. Press the #1 key.
13. Press ENTER.
14. Press F1.
15. Press the #3 key. This will enter the GO command. You will see the display shown here:

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 _?	
YES	NO

To Edit an Existing Program:

Follow the same steps as in "To Create a New Program", 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 Æ) scroll through programs one character at a time.
- The up and down arrows (≠ or Ø) scroll through programs one line at a time.

Naming Programs

Any or all of the programs stored in the non-volatile memory of the SmartDrive can be given descriptive names in addition to the program number that the SmartDrive assigns it. Program names must be put inside 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 almost 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 a SmartDrive, 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 Characters Using 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 more descriptive
- Use operator messages or prompts
- Send messages over RS-232C
- Use commands not on the keypad, such as EA or " "

General Rules for Using the ALPHA key

- Any letter and character located above a number on a numeric key may be inserted into a program, i.e. the A, B, or C above #1 on the #1 key.
- Press a numeric key 4, 5, and 6 times to access the lower case letters.
- You must press ALPHA prior to each character you wish to enter.
- Press the ¨ or Æ arrow key to move the cursor to the next space.
- Press ALPHA 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.

Entering a Character (those found above numbers on Keypad)

Example: to insert the A, B, or C on the #1 key:

1. Press ALPHA.
2. Press the #1 key once to enter A, press it twice to enter B, or press it three times to enter the C.

Using ↑ ↓ Arrow Keys for Additional ALPHA Characters

The 19 special characters shown to the right are available by pressing ALPHA and scrolling through the list using the arrow keys.

<	>	?	!	@	#	%	&	_	|
:	;	\	'	"	↑	↓	←	→	

1. Press ALPHA.
2. Press ↑ or ↓ to scroll through the list until the desired character is displayed.
3. When you find the desired character press ALPHA or ENTER to insert the character. The character will be displayed and the cursor will move one space to the right.
4. Scroll through the list to select your next character, or press ESC to leave the editor.

Example of Naming a Program

Add the name [MINE] to the program created earlier (AC.3 VE2 DI1 GO).

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.
12. Press ESC. You will be prompted as shown.

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-Menu	Setup Parameter	Description of Setup Parameter
MOTOR	TYPE	Motor parameters
	D-RES	Drive resolution
	DIR	Direction of travel
ENC	MODE	Select open/closed loop mode
	E-RES	Encoder resolution
	FOL-ERR	Following error
	IN-RANGE	Position maintenance window
	PMGAIN	Position maintenance gain
	PMMAX	Position maintenance maximum velocity
MECH	DIST	Distance Units
	RATIO	Scale distance to preferred user units
	BKLASH	Electronic backlash compensation (not implemented)
	VEL	Speed units
	VMAX	Critical speed limit
	ACCEL	Acceleration units
	AMAX	Maximum rate of acceleration/deceleration
I/O	INPUTS	Input functions
	OUTPUTS	Output functions
	OPTOS	OPTO module configuration
	OUTSTS	State of output on Power-up, Fault, or Stop
	LIMITS	End-of-Travel Switch Polarity
JOG	ACCEL	Jog acceleration
	LO-VEL	Low jog velocity
	HI-VEL	High jog velocity
	ENABLE	Enable/disable jog in RUN menu
HOME	MODE	Homing method
	EDGE	Edge of home switch
	SWITCH	Type of home switch
	OFFSET	Position counter offset
	DIR	Final homing direction (positive or negative)
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	DISP	Format Display
	STOP-RATE	Decel rate when stop input activated
	TEST	Enable Test Menu (not currently implemented)
	FAULT	Polarity - Fixed Active High in SmartStep
	ENABLE	Polarity - Fixed Active Low in SmartStep
	PASWRD	Password setup for operator/administrator access

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 SmartDrive is 6K bytes (maximum single program is 1,024 bytes), and will store up to 199 programs. The 30K memory option will store up to 400 programs.
Note: one byte equals one character in a program and on the keypad display.

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 Necessary only on brushless-servo Smart Drives

Using HELP

If you have a question while using the keypad, pressing HELP will display a help message related to the currently active menu. 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 ...

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. Command syntaxes and details on using commands are available in the Chapter 6 - *Programming Commands*, or from HELP in the Application Developer editor.

Note: A program must be selected to view the COMMAND SUMMARY.

Using COPY

Copying programs from one name (or number) to another, and downloading between keypads or between keypad and PC can save a significant amount of time when programming 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
```

PROG

PROG allows you to copy any existing program to a new program name.

```
↑ SOURCE PROGRAM ↓
> 5
```

To copy one program to another:

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.

Note regarding keypad and SmartDrive memory: standard memory on both the keypad and SmartDrive is 6K. The keypad cannot copy more than 6K of programs from a SmartDrive equipped with the 30K memory option.

TO PAD

Allows you to copy programs from the SmartDrive to the keypad or from a PC to the keypad.

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

To copy programs from the SmartDrive to the keypad:

1. Press F2 (TO PAD). The two messages to the right will appear sequentially on the keypad display.

   When the messages disappear, the programs have been downloaded to the keypad.

```
Receiving From Drive
```

```
Saving to EEPROM
```

To copy programs from a PC to the keypad:

1. Connect keypad to Computer using RS232 cable (IDC P/N PCS-5004).
2. In *Application Developer* **Communications** menu, click on **Send All** and choose **To Keypad**. The keypad will display the message Receiving From PC and a few more messages will quickly appear, then disappear from the screen.

   When the keypad display goes completely blank, the PC-to-keypad download is complete.

FROM

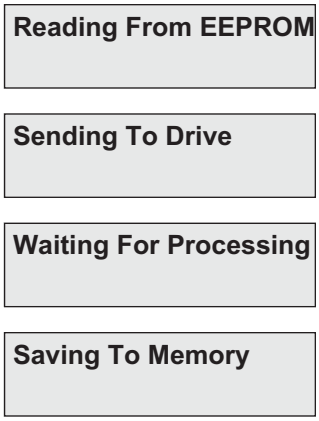
Provides a means of copying (sending) programs *from* the keypad to a SmartDrive or *from* the keypad to a PC.

To copy programs from keypad to SmartDrive

1. Press FROM (F3).

   The four messages to the right will appear sequentially on the keypad display.

   When the messages disappear, the download is complete.



To copy programs from the keypad to a PC

1. Connect keypad to Computer using RS232 cable (IDC P/N PCS-5004).
2. In *Application Developer* **Communications** menu, click on **Retrieve All** and choose **From Keypad**. The keypad will display the message Sending to PC and a few more messages will quickly appear, then disappear from the screen.

   When the keypad display goes blank, the keypad-to-PC upload is complete.

Using DELETE (DEL)

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

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 ↓
> _
```

To Delete a single entry (letter or number)

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

Chapter 5 - Configuring Your System

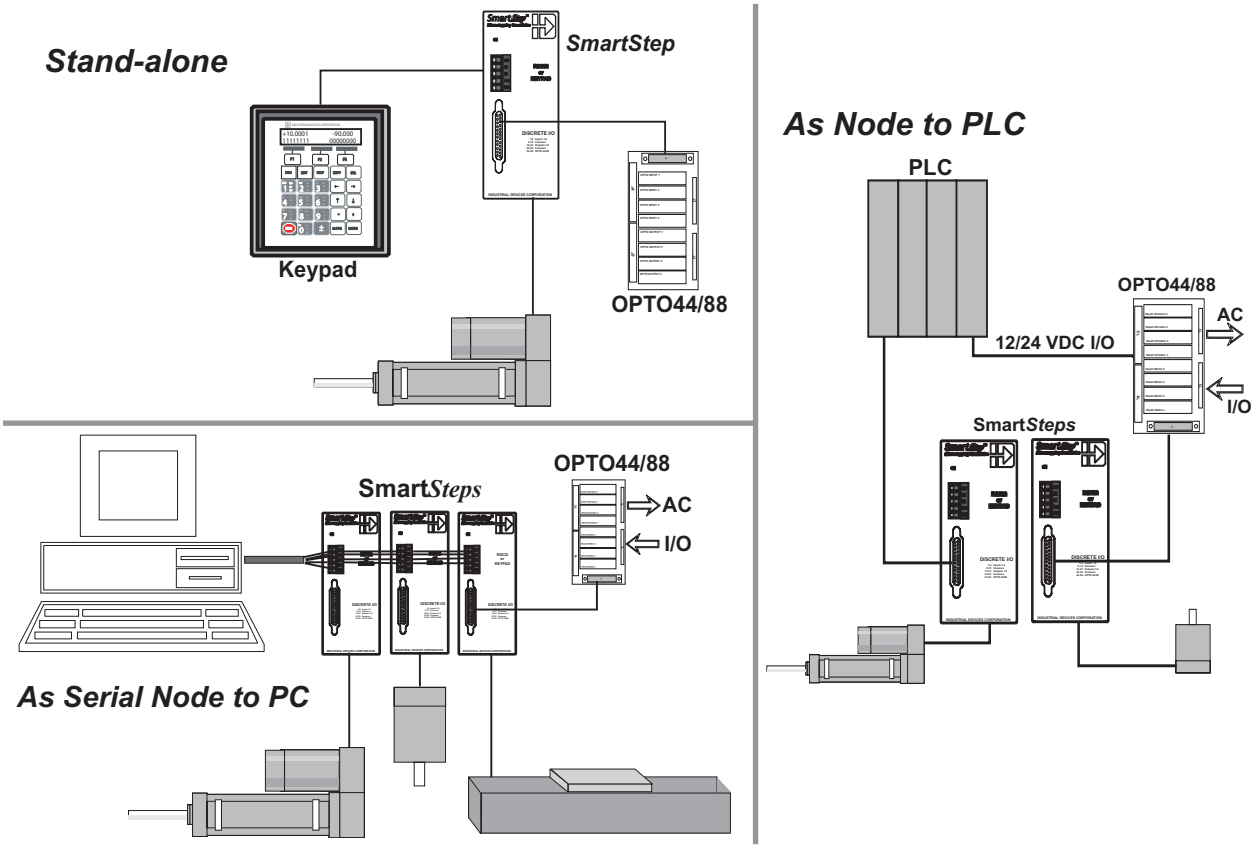
This chapter provides procedures for configuring your SmartStep to your specific equipment and application requirements. SmartStep configuration is divided into two categories. First is "Connecting Your Hardware" which is preceded by IDC's recommended wiring practices. Hardware connection is accomplished also by referring to Chapter 9, Hardware Reference. The second category is "Configuring Your System" which is the software setup of your application. This includes setting motor current, scaling units, I/O configuration, and more.

All of the software configuration can be done via the keypad, or via serial communication using *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.

*S*erial communication users should refer to this chapter for detailed explanations of configuration parameters. Details on using the *Application Developer* can be found in Chapter 8 - *Programming with Serial Communication*. For PC terminal users, non-Windows PC users, or PLC users, the equivalent 2-character ASCII configuration commands are detailed in Chapter 8. In this chapter, the 2-character ASCII command appears in brackets next to the keypad command. *Application Developer* users will find that the Windows dialog boxes follow the keypad menu structure very closely.

Typical SmartStep Application

The SmartStep easily interfaces with a PC, PLC, the Keypad, or can be used as a stand-alone machine controller. Block diagrams of several SmartStep-based applications are shown below:



Recommended Wiring Practices for IDC Controls

When configuring your SmartStep please follow the wiring practices listed below:

- Earth ground your machine at one point using a star configuration. Multiple earth grounds can cause a ground loop (see Preventing Ground Loop diagram below).

•Avoid long cable runs. The longer the cables, the lower the signal-to-noise ratio in your application.

•Use shielded motor and encoder cables along the entire cable run.

•Separate the signal wires (I/O, encoder, etc.) from the motor wires, AC power wires, and other sources of noise in your application.

•Avoid extending cables in the field via junction boxes, terminal strips, or Molex connectors. These types of connectors are typically unshielded (as is wire going into and out of the connector) and are places where noise may be injected into the system.

•Connect cable shields at the control end of your machine. Connect the motor cable shield to GND, and connect the encoder shield to COM on the control.

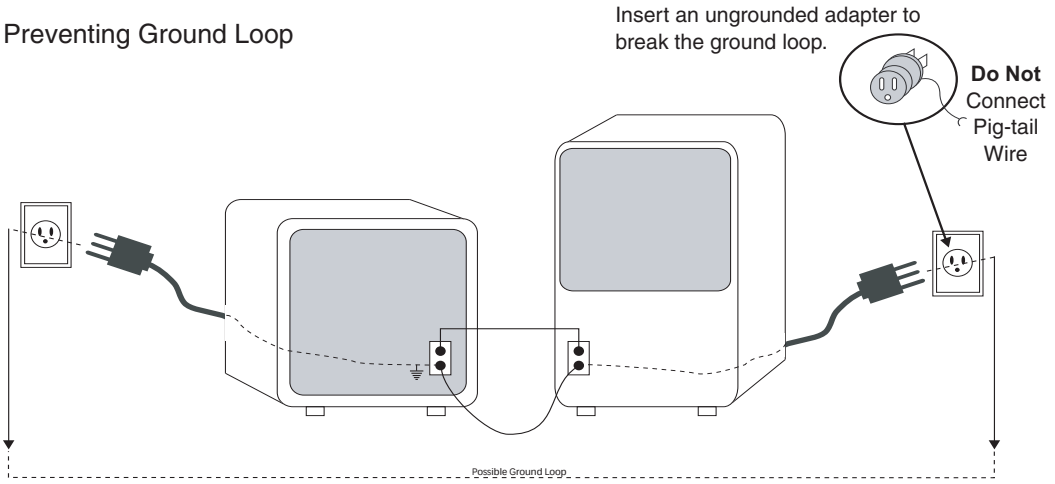
•Do not connect the logic common (COM) of the IDC control to earth ground on your machine or to the GND terminal on the control/drive. Separating earth ground from logic common minimizes the potential for ground loops.

•Use shielded cables inside your panel (if control is panel or cabinet-mounted) for I/O and encoder wiring.

•Use differential, line-driven encoders with shielded, twisted-pair encoder cables. Single-ended TTL encoders are susceptible to noise and should be avoided.

•For optimum noise immunity use IDC motors and encoders with IDC controls.

Preventing Ground Loop



Additional Wiring Practices When Connecting an IDC Control to a PLC

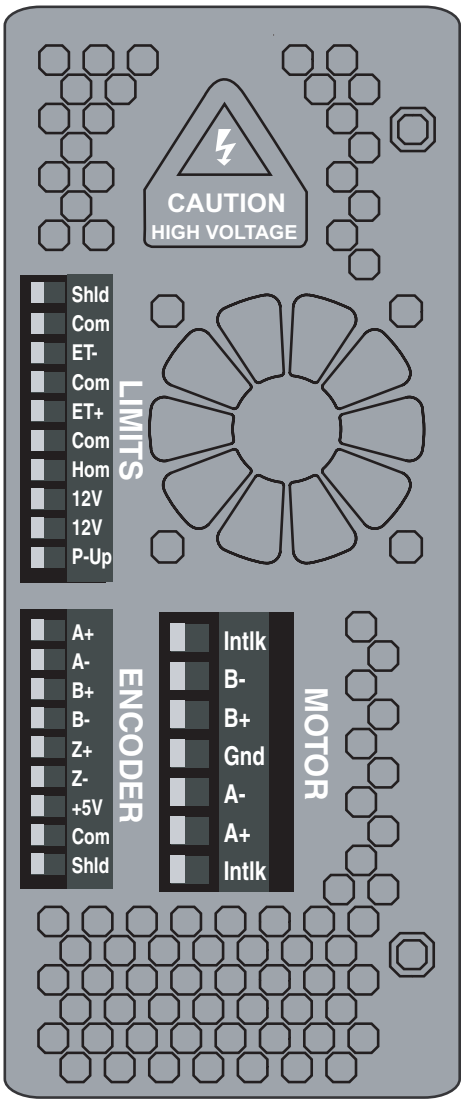
- Connect the PLC logic common and the IDC COM terminals together.
- Disconnect the jumper between Pull-Up (or P-Up) and +12 VDC on the IDC control. Connect the positive terminal of the PLC power supply to Pull-Up (or P-Up), and connect the Power Supply common to the COM terminal on the IDC control, or to the PLC logic common (these should both be at the same potential).
- IDC inputs are Sourcing, so PLC ouputs connected to IDC inputs should be Sinking.
- IDC outputs are Sinking, so PLC inputs connected to IDC ouputs should be Sourcing.

Connecting Your Hardware

1. Motor Wiring

The **A+, A-, B+** and **B-** phase outputs power the motor windings.

The two **Intlk** pins must be jumpered together at each motor connector to enable the drive to apply power to the motor. If an interlock wire breaks, or the connector is removed, the current to the motor is immediately stopped and the drive will fault (latch). Extending the interlock wire beyond 5 inches can lead to noise-generated shutdowns. Note: this is a low-impedence safety interlock circuit.



Gnd is an earth ground, internally connected to the power connector earth ground and to the control's chassis ground. This provides a convenient terminal for grounding the motor frame and a motor cable shield.

IDC Motors

Refer to the motor data sheets in Chapter 9, *Hardware Reference* for wiring IDC motors with Quick Disconnect cables.

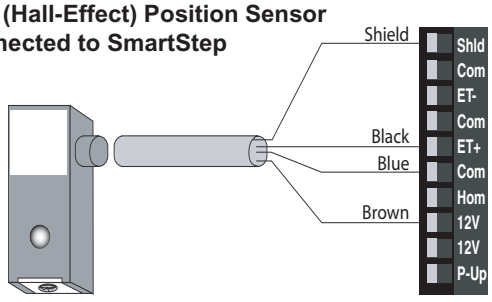
Connecting Non-IDC Motors

Refer to Non-IDC Motor section in Chapter 9, *Hardware Reference*.

2. Limits Connections

A typical IDC position sensor is shown connected to a SmartStep in the illustration below. Refer to pages 9-7 and 9-8 for more details on "Limits" connections.

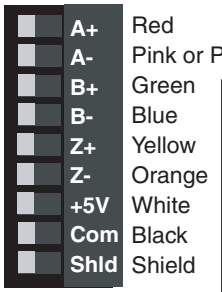
PSN (Hall-Effect) Position Sensor Connected to SmartStep



3. Encoder Wiring

An optional encoder port is available to allow for stall detection, closed-loop positioning, and position maintenance. Encoders are also used to position to the actual load position rather than a motor position. IDC recommends an encoder resolution of 8000 pulses per revolution or less with the SmartStep to prevent the possibility of end-of-move dither, which is caused by encoder positioning that falls between two motor step positions.

The color codes shown to the right apply to IDC supplied encoders only. Use the signal names to connect other manufacturer's encoders.



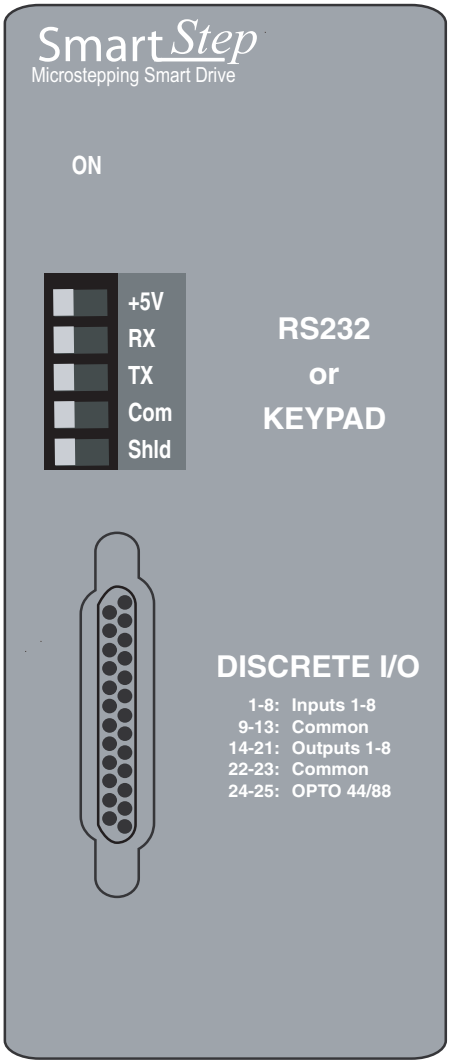
A+	Red
A-	Pink or Purple
B+	Green
B-	Blue
Z+	Yellow
Z-	Orange
+5V	White
Com	Black
Shld	Shield

Connecting an Encoder to a SmartStep

Color codes are for IDC encoders only. Use signal names for other manufacturer's encoders.

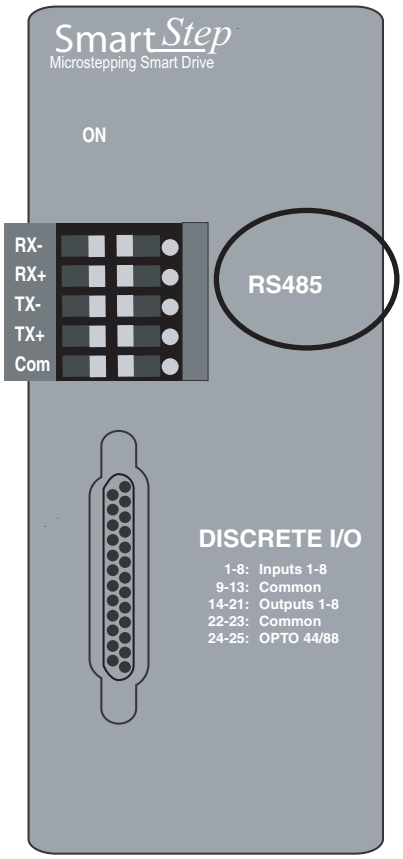
4. Connecting Your Keypad

Your FP220 Keypad will work normally with the standard SmartStep as shown to the right. Connect the keypad to the RS232 or KEYPAD connector on the front of the SmartStep.



WARNING: Your keypad will not work with the RS-485 version of the SmartStep.

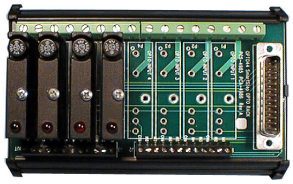
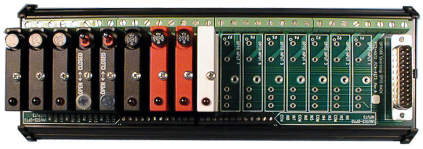
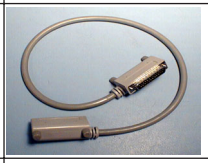
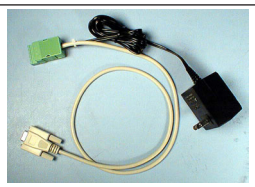
Do Not Plug Your Keypad into the RS-485 SmartStep (shown below).



5. Configuring Your Inputs & Outputs (I/O)

Your SmartStep has eight optically-isolated inputs, and eight discrete, optically-isolated outputs which may be configured to specific machine control functions (see page 5-22 for details). Unlike other IDC SmartDrives, the SmartStep does not have OPTO I/O positions. However, the SmartStep does have provisions for convenient connections and conditioning of machine I/O. A variety of SmartStep I/O accessories (shown below) are available to suit your application requirements.

SmartStep I/O Accessories

Accessory (P/N)		Description
OPTO44		OPTO Rack that accepts up to 8 optional conditioning modules. (See Opto Module in table in Chapter 9, Hardware Reference, for list of modules available from IDC).
OPTO88		OPTO Rack that accepts up to 16 optional conditioning modules. (See Opto Module table in Chapter 9, Hardware Reference, for list of modules available from IDC).
DB25BO		Screw Terminal Breakout Board. See connection information in Chapter 9.
SS-PNP-BO		Screw Terminal Breakout Board to Convert to Sourcing Outputs. See connection information in Chapter 9.
SS-IO SS-IO-6		I/O cables that connect SmartStep to other devices or a PLC. SS-I/O is 2 ft. SS-I/O-6 is 6 ft.
PCS-4991		Cable for connecting SmartStep to PC (9-pin Comm. Port).
PCS-5004		PC-Keypad Cable for copying programs between keypad and PC.

For more information on how to use your SmartStep's inputs and outputs in an application, refer to *Chapter 6*, *Programming Commands,* and *Chapter 7*, *Programming Your Application, and Chapter 9, Hardware Reference*.

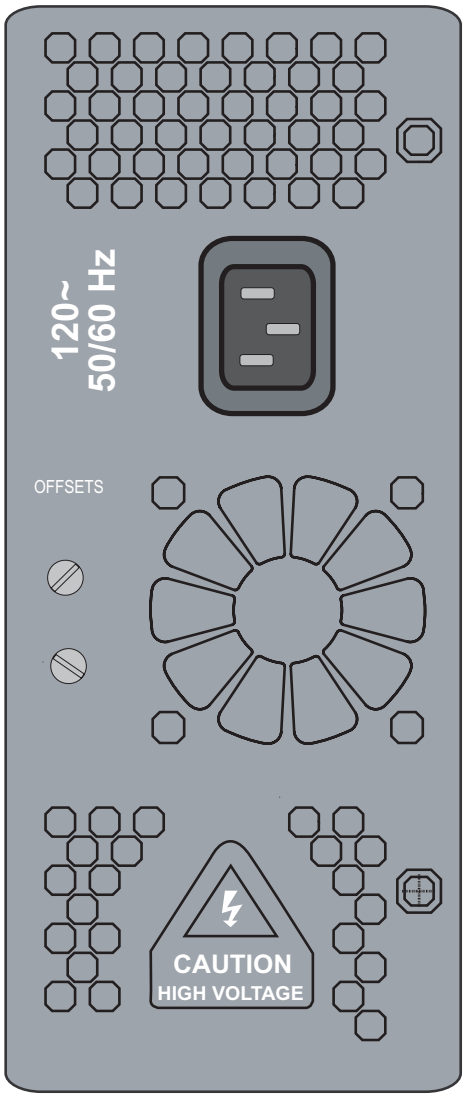
6. AC Power

AC power is plugged into the connector on top of the SmartStep.

SmartStep input voltage must be in the range of 90 - 120 VAC single phase, 50/60Hz, 500 VA max. @ 7.9 amp setting.

SmartStep240 input voltage must be in the range of 190 - 240 VAC single phase, 50/60 Hz, 500 VA max @ 4.0 amp setting.

SmartStep23 input voltage must be in the range of 90 - 120 VAC single phase, 50/60 Hz, 250 VA max @ 3.0 amp setting.



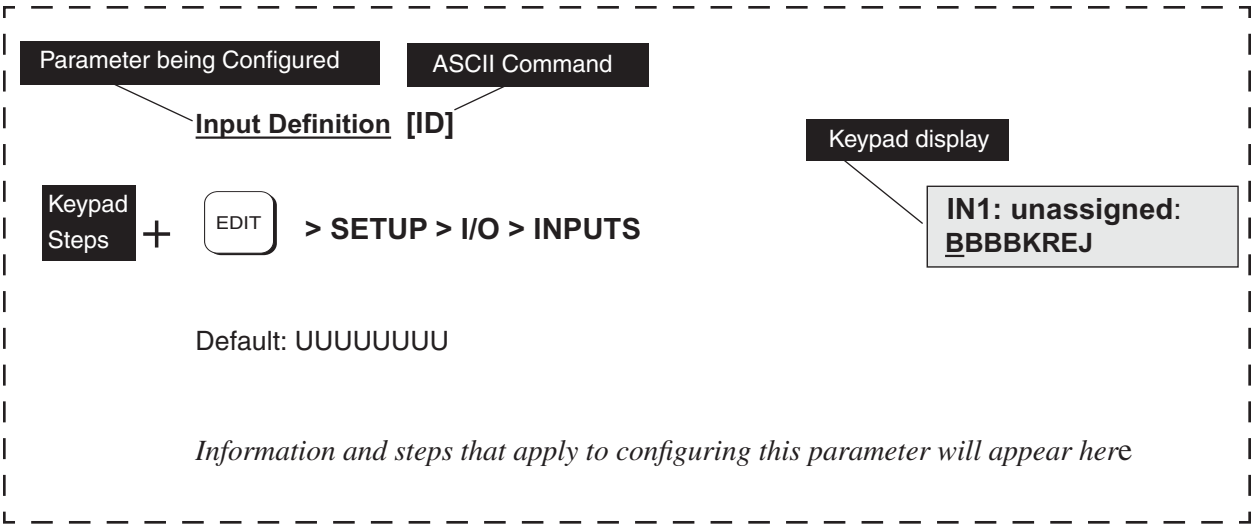
Configuring Your System

This chapter contains details and directions for customizing the SmartStep 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.

Application Developer and serial communication users should refer to this chapter for detailed explanations of configuration parameters. In this chapter, the 2-character ASCII command appears in brackets next to the keypad command. *Application Developer* users will find that the Windows dialog boxes follow the keypad menu structure very closely. PC, non-Windows PC, or PLC users will find details on using IDC's *Application Developer* in *Chapter 8 - Programming with Serial Communication*.

The task of configuring your SmartStep to a specific application consists of customizing a number of software parameters to match the mechanics of the system. These parameters include motor setup, encoder, distance, acceleration and velocity scaling, I/O, jog, home, and serial communication.

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 three parameters as shown at top right.

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

Press ↑ or ↓ to access the remaining six parameters.

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

Configuring your system with the keypad begins below:

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

Configuring Your Stepper Motor

Adjustments for **Current, Waveform, Rest, Idle, Inductance, and Anti-Resonance** can be made while the motor is energized and moving. Motor References 1 and 2 on the following page will help you accurately configure your stepper motor.

Configuring Motor Type [MT11]

Note: When you select MOTOR > TYPE, you will see three motor choice; STEPER, R-SRVO, AND L-SRVO. The SmartStep will only allow you to configure the STEPER option. The other options are for configuring servo motors with the versatile keypad.

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

```
-↑ STEPPER SETUP-↓:
CURRENT A-RES INDUCT
```

Configuring Motor Current [MIn]

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

Default: 0.0 Amps
Range: 0.0 - 8.0 Amps for SmartStep
0.0 - 3.0 Amps for SmartStep23
0.0 - 4.0 Amps for SmartStep-240

The Motor Current parameter sets the motor current for your stepper motor. Entering a current outside the valid range will reset the motor current to 0.0 Amps.

1. Select CURRENT (F1). Default current is 0.0 Amps.

```
Axis One Motor Curnt
----Amps
```

2. From the IDC MOTOR REFERENCE table, enter the current that corresponds with the motor you are configuring.

3. Press ENTER.

Motor Reference - 1									
IDC MOTOR **SERIES (T)**	CURRENT @120	CURRENT @240	Inductance Setting	Unloaded Anti-Res	IDC MOTOR **PARALLEL(V)**	CURRENT @120	CURRENT @240	Inductance Setting	Unloaded Anti-Res
S12	1.0	1.0	LOW	29	S12	2.0		LOW	29
S21	1.2	1.2	HIGH	30	S21	2.3		LOW	27
S22	1.5	1.5	HIGH	28	S22	3.0		LOW	24
S23	1.7	1.7	HIGH	25	S23	3.4		LOW	22
S32	2.8	2.8	HIGH	22	S32	5.6		LOW	18
S33	3.5	3.5	HIGH	21	S33	7.0		LOW	17
S42	6.0	4.0	LOW	16	S42	7.9		LOW	12
P21		0.7	HIGH	30	P21	1.3		LOW	27
P22		1.0	HIGH	28	P22	2.0		LOW	24
P31		1.5	HIGH	27	P31	2.9		HIGH	23
P32		1.6	HIGH	24	P32	3.3		HIGH	20
P33		2.0	HIGH	22	P33	4.0		HIGH	18
P41		2.8	HIGH	21	P41	5.7		HIGH	17
P42		3.3	HIGH	18	P42	6.6		LOW	17
P43		3.3	HIGH	15	P43	6.6		HIGH	14
K31		1.5	HIGH	24	K31	2.9		HIGH	20
K32		1.7	HIGH	22	K32	3.4		HIGH	18
K33		1.7	HIGH	20	K33	3.3		HIGH	16
K41		2.8	HIGH	18	K41	5.7		HIGH	14
K42		3.2	HIGH	17	K42	6.4		HIGH	13
K43		3.3	HIGH	15	K43	6.6		HIGH	11

Motor Reference - 2															
MOTOR	S21	S22	S23	S32	S33	S42	P21	P22	P31	P32	P33	P41	P42	K42	K43
OFFSET TEST SPEED	4.46 RPS	3.91	3.57	2.98	2.76	2.11	3.78	4.06	3.54	3.49	3.36	2.85	2.75	3.14	3.75
WAVEFORM TEST SPEED	1.12 RPS	.98	.89	.74	.69	0.53	0.95	1.01	.88	.87	.84	.71	.69	.78	.94
J_{rotor} (INERTIA)	1.17E-5 $kg\text{-}m^2$	2.34E-5	3.51E-5	1.21E-4	1.88E-4	8.0E-4	2.48E-5	4.31E-5	1.40E-4	2.70E-4	4.00E-4	5.50E-4	1.09E-3	1.09E-3	1.62E-3

Configuring Anti-Resonance [ARi]

EDIT **> SETUP > MOTOR > TYPE > STEPER > A-RES**

Default: 0
Range: 0 - 30

The Anti-Resonance parameter sets the anti-resonance gain level for your motor. For example, issuing an AR14 would set the anti-resonance gain to 14.

1. Press A-RES.
2. At this point, you have three options:

-Axis One Anti-Res-
0

Option 1: Enter the "Unloaded Anti-Res" setting for your IDC motor (see Motor Reference 1). Skip to step 4.

Option 2: Calculate an exact AR setting for your IDC motor (see Calculating AR below, and Motor References 1 and 2). Go to step 3.

Option 3: If you are configuring a non-IDC motor, turn to "Non-IDC Electric Motors", "Calculating $\mathbf{AR_{unloaded}}$" at the end of Chapter 9. Calculate $\mathbf{AR_{unloaded}}$ and go to step 3 if your motor is unloaded. If an exact AR value is required, use your $\mathbf{AR_{unloaded}}$ value to calculate AR (see formula below). Proceed to step 3.

Calculating AR

$\mathbf{AR = AR_{unloaded} - K}$

$\mathbf{AR_{unloaded}}$ is found in Motor Reference 1; **K** must be calculated using the following formulas:

$$\mathbf{K = \frac{Log(N)}{.155}} \qquad \mathbf{N = \frac{J_{rotor} + J_{load}}{J_{rotor}}}$$

$\mathbf{J_{rotor}}$ is found in Motor Reference 2

$\mathbf{J_{load}}$ is customer supplied

3. Enter your Anti-Res value (this will be a number between 0 and 30).
4. Press ENTER.

NOTE: Empirically, the AR value will decrease as the $\mathbf{J_{load}}$ increases. You can observe this by entering smaller AR settings until the motor begins to hiss - then increasing your AR setting slightly.

5. Press ESC to return to the STEPPER SETUP menu.

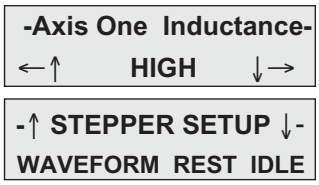
Configuring Motor Inductance [MHa]

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

Default: HIGH

The SmartStep works best with motors higher than 4 mH. The Motor Inductance parameter configures the drive for a high or low inductance motor. Motors above 10mH are considered HIGH for SmartStep and SmartStep 23. Motors above 40mH are considered HIGH for SmartStep-240.

1. Press INDUCT.
2. Using Motor Reference - 1, select HIGH or LOW Inductance using ↑↓ keys.
3. Press ESC to return to the STEPPER SETUP menu.



Configuring the Waveform [WAi]

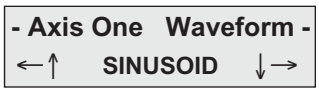
EDIT **> SETUP > MOTOR > TYPE > STEPER > WAVEFRM**

Default: SINUSOID

The Waveform parameter configures the SmartStep for either a pure sinusoid waveform or a -4% 3rd harmonic waveform.

Depending on motor design and the current level at which it is being driven, it may be advantageous to distort the sinusoidal waveform to achieve better low speed smoothness and step-to-step accuracy. With skewed rotors or 50-48 tooth geometry pure sine will usually produce the best results.

1. Press WAVEFRM (F-1). Waveform is configured only once per motor model for better low speed smoothness.
2. With the motor running at the speed indicated in the WAVEFORM TEST SPEED row (see previous table), alternately select between SINUSOID and -4% 3rd to determine which setting produces the smoothest running condition.
3. Press ESC to return to STEPPER SETUP.



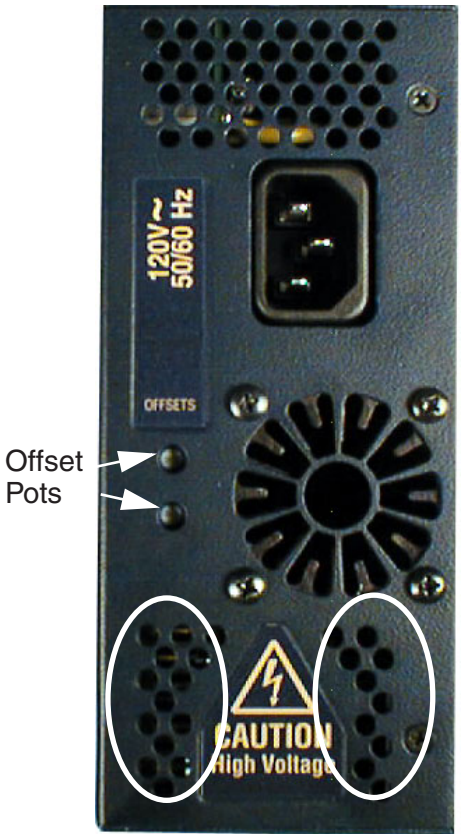
Fine-Tuning Offsets

These potentiometers adjust the phase offset between Phase A and Phase B.

To adjust Offsets:

1. Run the motor unloaded at the Offset Test Speed indicated in Motor Reference 2.
2. Alternately adjust the pots for smoothest running condition.

WARNING! - The Offset potentiometers (see photo) are located near other holes in the SmartStep housing. **Do not insert your screwdriver in the holes adjacent to the CAUTION label! You could be injured by electrical shock**. See the circled holes in the illustration on the right.



Configuring Rest Mode [REi]

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

Default: OFF

When Rest Mode is enabled, motor current is reduced to 1 Amp if no motion occurs for 12 minutes. Full current is restored when the next move starts. Enabling REST reduces motor heating and improves fan life in applications where the machine is powered-up but may not run for extended periods of time, e.g. a machine that is operated during two shifts, but is left on 24 hours a day.

1. Use arrows (↑↓) to select ON or OFF.
2. Press ESC to return to STEPPER SETUP.

-Axis One Rest Mode-
←↑ OFF ↓→

Configuring Idle Mode [ILi]

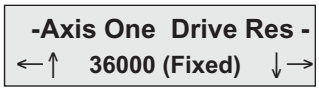
EDIT **> SETUP > MOTOR > TYPE > STEPER > REST**

Default: OFF
When Idle is enabled, the motor current is reduced to 75% of the programmed value if no motion occurs for 10ms. Full programmed current is restored on the first pulse of the next move.

1. Press IDLE.
2. Use arrows (↑↓) to select ON or OFF.

Configuring Drive Resolution [MR10]

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

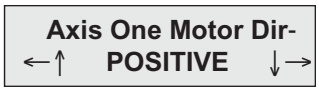


Default: 36000 steps/rev (fixed)

1. The Drive Resolution is fixed at 36,000 as shown in the display example above.

Configuring Motor Direction [MDi]

EDIT **> SETUP > MOTOR > DIR**



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 the ↑ ↓ and ENTER keys to select a direction

Configuring Your Encoder

If you are not using an encoder, only the encoder mode must be configured. **Ensure that OPEN LOOP is selected if you are not using an encoder,** and skip to **Configuring Your Mechanics.**

Configuring Encoder Mode [EMi]

EDIT **> SETUP > ENC > MODE**

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

Default: OPEN LOOP

This option sets the encoder mode. Encoder feedback is strictly optional with the SmartStep.

1. 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, and 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 actual encoder position, but moves are based only on commanded OPEN LOOP position. Stalls are detected in this mode.
CLOSED LOOP-PM	Functionally identical to CLOSED LOOP, with the addition of post-move position maintenance of the last commanded position. Provides “pseudo-servoing” to stepper systems. Use PM GAIN, PM VMAX, and IN-RANGE WINDOW setup parameters to specify position maintenance tuning parameters. Application Notes: Following-error is still active while in CLOSED LOOP-PM mode. A following-error will occur when the number of correction steps exceeds the following error value. This allows the unit to signal a fault when the displacement cannot be corrected, i.e. an obstruction. CLOSED LOOP-PM will not attempt to correct position while navigating menus with the keypad.

Configuring Encoder Resolution [ERi]

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

```
Axis One Encoder Res -
←↑      2000      ↓→
```

Default: 2,000 pulses per rev.

This option is used to set the encoder resolution. The resolution is specified in encoder pulses per revolution of the motor, post-quadrature. To prevent end-of-move dither with a SmartStep, we recommend an encoder resolution of 8000 pulses per revolution or less.

1. Use the numeric keys to enter the encoder resolution.

Configuring Following Error Limit [FEi]

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

```
- - Axis One Fol Error --
←↑   750  steps   ↓→
```

Default: 750 motor steps
Range: 0-999,999 motor-step counts, 0 = OFF
*Units: 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.

1. Use the numeric keys to enter the Following Error limit.

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 serial command, the ESC key or a RESET will clear a Following Error fault
- Bit 9 of SS response is set to 1
- Bit 1 of SD response is set to 1

* Following Error is always set in *motor steps*, not encoder steps.

Configuring Position Maintenance In-Range Deadband [IRi]

EDIT **> SETUP > ENC > IN-RNGE**

```
- IN-RANGE SETUP-
WINDOW  TIME
```

Default: 25 encoder steps

In-Range Window specifies the position maintenance deadband or region surrounding the set-point position. The "window" is specified in post-quadrature (4 x # of lines) encoder steps.

The window is the region surrounding the commanded position in which the motor shaft can reside and not be considered "out of position." The control will try to correct the position if the motor is outside this window.

1. Select WINDOW (F1) in the IN-RANGE SETUP menu.
2. Enter the desired number of encoder steps (must be a positive number).
3. Press ENTER.

Configuring Position Maintenance Gain [PGi]

EDIT **> SETUP > ENC > PMGAIN**

```
- Axis One PM Gain- -
←        10        →
```

Default: 10
Range: 1 - 32,767

PM Gain specifies a gain value used to determine correction velocity. The correction velocity is calculated as "displacement* correction gain" in units of steps/sec. Therefore, the larger the displacement, the faster position maintenance will attempt to correct position. For example, if the correction gain is set to 3 and an active displacement of 3200 steps occurs, the correction velocity will be (3 * 3200) = 9600 steps/sec.

Configuring Position Maintenance Max Velocity [PVi]

EDIT **> SETUP > ENC > PMMAX**

```
- Axis One PM MaxVel-
←      1.0 rps      →
```

Default: 1.0 rps
Range: 0.005 - 9,999,999

PMMAX limits the velocity of a position maintenance correction. Regardless of the magnitude of displacement or correction gain, the correction velocity will never exceed the maximum velocity setting.

Configuring Your Mechanics

Through the MECH SETUP menu, your SmartStep 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.

```
←↑ MECH SETUP ↓→
DIST   RATIO   BKLASH
```

```
←↑ MECH SETUP ↓→
VEL    VMAX    ACCEL
```

Pressing MECH displays three menu choices:

Pressing ↓ or↑ reveals four additional menu choices:

```
  ↑ MECH SETUP ↓
AMAX
```

Configuring the Distance Unit [DUi]

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.

Use the↑ ↓ keys to select distance units from the following list:

mils	arcmin	inch	degrees	µm
feet	radians	yards	grads	
steps	cm	%	meter	
arcsec	revs	mm	index	

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 Ratio (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 [GRi:i]

EDIT **> SETUP > MECH > RATIO**

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

Default: 1 to 1

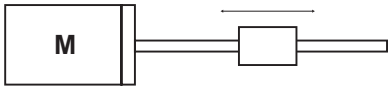
The RATIO option is used to scale DI and DA moves to your preferred distance units. RATIO sets the ratio of *motor revolutions per DIST unit*. Up to 5 digits on either side of the ratio can be entered to properly scale your DIST units. Make certain that the RATIO accurately represents the Distance Unit (DU).

1. Use 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 0.2"

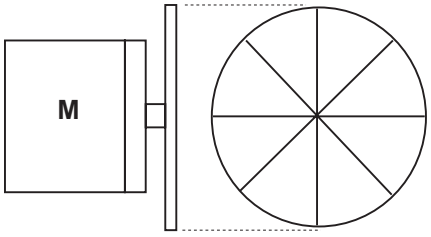
Notes:
You can change DIST or RATIO at any time. Changing them will not change the associated DI or DA values in a program, so all moves will change by the same factor that RATIO was changed.

If using an IDC supplied actuator, the proper Gear Ratios for entering units of *Inches* and *mm* can be found in Appendix A.

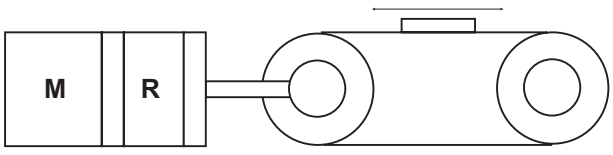
Units Example - Lead Screw System
- Desired distance units: inches
- Leadscrew: 4 revs/inch
- DIST = inch
- RATIO = 4 to 1



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
- DIST = mm
- RATIO = 50 to 1524



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.

Configuring the Units of Velocity [VUi]

EDIT **> SETUP > MECH > VEL**

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

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 the ↑↓ and ENTER keys to select velocity units from the list:
   - rps (motor velocity not affected by Gear Ratio)
   - rpm (motor velocity not affected by Gear Ratio)
   - (DIST units)/sec (see SETUP>MECH>DIST)
   - (DIST units)/min (see SETUP>MECH>DIST)

Configuring Maximum Velocity [MVr]

EDIT **> SETUP > MECH > VMAX**

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

Default: 50 {velocity units}

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

1. Use the numeric keys to set the maximum velocity in VEL units.

Configuring Acceleration Units [AUi]

EDIT **> SETUP > MECH > ACCEL**

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

Default: sec

This option is used to select acceleration (and deceleration) units. All acceleration and deceleration values specified in the system will be expressed in these units. You may specify acceleration as a rate, or in time-to-accelerate to full speed. Motor shaft acceleration is not affected by Gear Ratio.

1. 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/sec^2)

Configuring Acceleration Maximum [AMr]

EDIT **> SETUP > MECH > AMAX**

- Axis One Max Accel -
← 0.002 sec →

Default: 0.002 seconds or 999999 units/sec^2

Acceleration Maximum command sets a maximum acceleration and deceleration limit for programmed move profiles in the current acceleration units. Programmed accelerations and decelerations for moves will be limited by this parameter (analogous to VMAX for velocity).

1. Enter the desired Acceleration Maximum
2. Press ENTER.

Configuring Your Inputs & Outputs

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

```
- - - - I/O SETUP - - - -
INPUTS  OUTPUTS  OPTOS
```

Configuring Your Inputs

Configuring Input Definition [IDaaaaaaaa]

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

```
IN1:UNASSIGNED
UUUUUUUU      ←↑↓→
```

Default: UUUUUUUU

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.

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:

Note: *e, f, j, m, and r* (lower case) are on the keypad but are not used with SmartStep.

Input Characters and Keypad Display

Character	Keypad Display
B	Bin Program
C	BCD Program
c	Clear Command Buffer
D	Lock Keypad
E	Extend Jog 1
*F	Set Force 1
I	Interrupt (RUN 98)
J	Jog Speed 1
K	Kill
M	Shutdown 1
N	Analog Input
P	Pause/Continue
R	Retract Jog 1
G	Registration
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 *255* 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.

c Clear Command Buffer
Clears the terminal input buffer and buffered command buffer

D Lock (Disable) Keypad
When activated, the keypad is disabled allowing NO user access. The keypad resumes normal operation, subject to the DIP switch pattern, when the input is released.

E Extend Jog (E specifies axis 1)
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, as will pressing the ESC key on the keypad.

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, Input #1 must be configured as a Registration input - 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 SmartStep 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. 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 and (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. Subsequent interrupts are ignored until (ARM INT98) is reset to 1.

(INT98CTRL) and (ARM INT98) are reset to default values on power-up. Note: There is a space in (ARM INT98).
When activated, any executing program or functional operation is terminated and program I98 (interrupt program) is immediately executed. If a move is executing when the interrupt is activated, the move is terminated (decelerated at a rate determined by the Stop Deceleration rate setup parameter). The unit will go into Run mode once program I98 is completed

J Jog Speed (J specifies axis 1)
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
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 Motor Shutdown: (M specifies axis 1)
Disables amplifier. 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
The SmartStep has analog input capabilities when used in conjunction with an OPTO 44 or OPTO 88 rack. Analog input configuration is limited to inputs #1 - #6 only. Analog input values (AIx - the Built-In Variable) are updated every 16ms.

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 Retract Jog (R specifies axis 1)
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 serial communication will also clear the fault, as will pressing the ESC key on the keypad.

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.

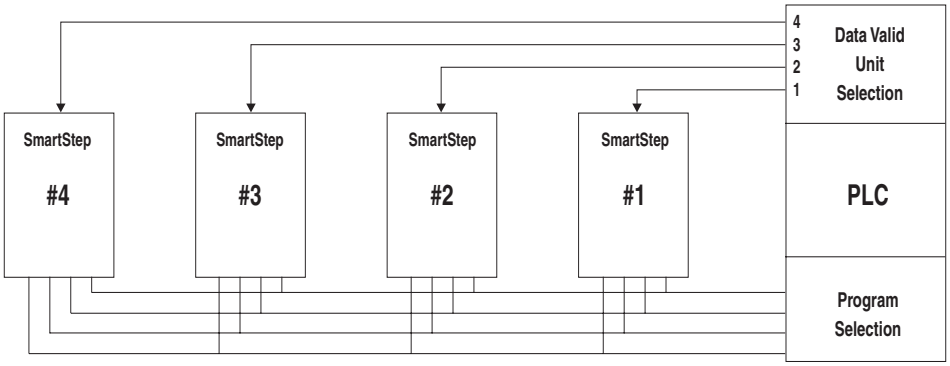
See the **SCAN** setup parameter for more information on stopping program execution. See the **ST** command in Chapter 3 for more information on stopping moves without halting command execution.

***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.

*Default configuration

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 SmartStep, 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 [ODaaaaaaaa]

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

```
OUT1: PROGRAMMABLE
PPPPPPPP←↑↓→
```

Default: PPPPPPPP

The function for each output channel is indicated by a letter along the bottom of the display. The first 8 letters are for the dedicated Outputs.

1. Use ← and → keys to select an 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.

Note: *b, d, h, k, and m* (lower case) are on the keypad but are not used with SmartStep.

Output Characters and Keypad Display	
Character	**Keypad Display**
A	Amp Fault
B	Brake 1
C	Over Current (not yet implemented)
D	Direction 1
F	Fault
H	At Home 1
***K**	At CI Limit 1
L	Limit Error
M	Move Done 1
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 Brake (B specifies axis 1)

CAUTION
IDC offers brakes for the actuator screw, or as an integral part of some motors. 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 discuss the issue of brakes with an IDC Applications Engineer or with your distributor. **Note:** "Brake Output" connection examples may be found on page 9-13 of the Hardware Reference chapter.

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, 120VAC power supply, or 24 VDC with B Motors.

C Over Current (not yet implemented)

D Direction (D specifies axis 1)
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:

- Keypad display
- Over RS-232C using the **SS**, **SD**, and **SA** status commands (see Chapter 8)
- Other outputs can be configured to show more specific fault states

H At Home (H specifies axis 1)
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 Move Complete (M specifies axis 1)

The output goes high when an axis move is started and goes low when a move is completed.

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 - n/a on SmartStep

Configuring Your Optional OPTO Modules

Opto Definition [OP]

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

Use OPTO 44/88 rack with SMARTSTEP

The SmartStep does not have onboard OPTO I/O. Use OPTO44 or OPTO88 module to condition your I/O. The I/O on these racks is not bidirectional, so configuration is not necessary.

Configuring Your Output States

Configuring Output States on Power Up [OEP]

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

On PwrUp Output #1
←↑ OFF ↓→

Default: OFF
Range: n/a

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

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

Configuring Output States on Fault [OEF]

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

On Fault Output #1
←↑ OFF ↓→

Default: NO CHANGE
Range: n/a

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

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

Configuring Output States on Stop / Kill [OES]

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

Default: NO CHANGE
Range: n/a

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

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

Configuring Your End-of-Travel Switch Polarity

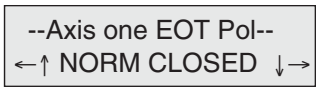
Configuring E-O-T Switch Polarity [ET]

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

Default: NORM CLOSED
Range: n/a

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

1. Use the ↑ and ↓ keys to select NORM OPEN or NORM CLOSED and press ESC to save your choice.



Configuring Your Jog Parameters

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

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

The parameters which control your jog operation are configured using the JOG SETUP menu:

```
↑ JOG SETUP ↓
ENABLE
```

Note: The Units used by the Jog parameters are configured from the SETUP > MECH menu.

Configuring Jog Acceleration [JAr]

EDIT **> SETUP > JOG > ACCEL**

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

Default: 0.3 {Accel Units}

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

1. Use 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 [JLr]

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

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

Default: 0.5 {Velocity Units}

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

1. Use 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 [JHr]

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

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

Default: 2.0 {Velocity Units}

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

1. Use 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 [JEi]

EDIT **> SETUP > JOG > ENABLE**

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

Default: Enabled

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

1. Use ↑↓ keys to enable and disable the function.

Configuring Your HOME Parameters

Your SmartStep 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.

↑ HOME SETUP ↓
EDGE LEVEL OFFSET

Configuring Home Edge [HEi]

EDIT **> SETUP > HOME > EDGE**

-Axis One Home Edge -
← NEGATIVE →

Default: NEGATIVE

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

1. Use ↑↓ keys to select the active edge as the positive or negative side of the home switch.

Configuring Home Switch [HSi]

EDIT **> SETUP > HOME > SWITCH**

Axis One Home Switch
← Norm Open →

Default: Norm Open

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

2. Use ↑↓ keys to select the switch type. (NORM OPEN or NORM CLOSED)

NOTE: To save inventory (part numbers) you may want to use only N.C. switches.

Configuring Home Offset [HOr]

EDIT **> SETUP > HOME > OFFSET**

```
Axis One Home Offset-
←  0.0      rev  →
```

Default: 0.0 {Distance Units}

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

1. 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 [HFi]

EDIT **> SETUP > HOME > FINAL**

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

Default: POSITIVE

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

1. Use ↑↓ keys to select the final approach direction.

Configuring Homing Mode [HMi]

EDIT **> SETUP > HOME > MODE**

```
--Axis One Home Mode
←↑  Switch Only  ↓→
```

Default: Switch Only

The Homing Mode parameter establishes how a Go Home (GH) command will execute homing routines. There are three modes of operation:

Switch Only	The control will only search for the appropriate edge of a switch.
***Switch Then Z Channel**	The control first looks for a home switch, aligns to the edge, and then slowly moves until an encoder Z pulse is found.
***Z Channel Only**	The control does not search for a home switch, instead it rotates at a slow speed until an encoder Z pulse is found, regardless of the home switch state. The magnitude of GH velocity parameter is ignored. The sign of the velocity parameter determines the low speed direction.

*Requires an encoder

Use ↑↓ keys to select the homing method.

Configuring Your PROGRAM SETUP Parameters

The Program Setup menu allows selection of (1) a program to be immediately run when the SmartStep is powered-up and (2) scanning conditions for the BCD or binary program select inputs.

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

Configuring Power-Up Program [PUi]

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

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

Range: n = 0 to 400
Default: 0

This option selects a power-up program. The selected program is executed (run) when your SmartStep 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.

1. Use numeric keys to enter a program number.

Configuring Scan Conditions [SNaaaaaaaa]

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 SmartStep 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). Each event is represented by one of seven Y/N positions on the bottom display line.

1. 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 ↑↓ keys to enable (Y) or disable (N) the selected event.

Configuring Scan Delay [DYi]

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 2 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.

1. Use the numeric keys to enter a value in ms.

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

Configuring Your Serial Communications

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

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

Fixed RS-232C parameters:
- Baud rate: 9600
- Data bits: 8
- Stop bits: 1
- Parity: none

Configuring Echo Enable [ECi]

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. **Note**: when a SmartStep is used to control a daisy chain as a "master" to several "slave" units, the ECHO must be **disabled** on the "master" unit.

1. Use the ↑↓ keys to enable or disable ECHO.

Configuring Unit Number [UNi]

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.

1. Use the numeric keys to enter the unit address.

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

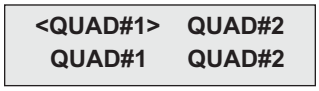
Configuring Your Miscellaneous Setup Parameters

The miscellaneous setup (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 Display Format [DF]

EDIT **> SETUP > MISC > DISP**

Default: Quad #1: POS1
Quad #2: BLANK on SmartStep
Quad #3: INPUTS
Quad #4: OUTPUTS



Range: n/a

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

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

Once a quadrant is selected, there are 9 possible data types that can be displayed in that quadrant.

Data Type	Quadrant Display
BLANK	No display
POS1	Axis position
POS1+UNIT	Axis position with axis units
VEL1	Axis commanded velocity
INPUTS	Discreet input status (0 off, 1 on)
OUTPUTS	Discreet output status (0 off, 1 on)
SA_STATUS1	Displays SA serial command response for axis
SS_STATUS	Displays SS serial command response
TEXT	Display user defined text in a quadrant

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

Configuring Stop Decel Rate [SRi]

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

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

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

This option is used to set the deceleration rate whenever a configurable stop input is activated, or when the ESC key is pressed while moving. This is usually set to the fastest controllable deceleration rate possible with mechanics in your application, or whenever a limit switch is hit.

1. Use the numeric keys to enter a stop deceleration.

Configuring Enable Line Polarity [EL0]

EDIT **> SETUP > MISC > ENABLE**

```
- Axis One EnablePol -
ACTIVE LOW
```

Default: ACTIVE LOW (Fixed Low on SmartStep)

This parameter is fixed in the SmartStep.

Configuring Fault Line Polarity [FL1]

EDIT **> SETUP > MISC > FAULT**

```
- Axis One FaultPol -
ACTIVE HIGH
```

Default: ACTIVE HIGH (Fixed High on SmartStep)

This parameter is fixed in the SmartStep.

Configuring Passwords [PWaaaa,aaaa]

EDIT **> SETUP > MISC > PASWRD**

```
- -PASSWORD SETUP--
OPRATR ADMIN CLEAR
```

Default: None

In addition to the keypad DIP switches, user-definable passwords also enable you to restrict access to the RUN, EDIT, COPY and DEL menus.

1. Enter your desired password, using the same keypad entry techniques described in Chapter 1. Use ←→ and DEL keys to edit the password.

2. Press ENTER to register the password.

SmartStep allows up to two types of passwords: **OPRATR** (Operator) and **ADMIN**.

Passwords and Menu Accessibility

This Password	Will Give You Access to These Menus
OPRATR only	RUN, EDIT, COPY, DEL
ADMIN only	RUN, EDIT, COPY, DEL
OPRATR + ADMIN	ADMIN = RUN, EDIT, COPY, DEL OPRATR = RUN only (All RUN functions except TEST)

General Password Rules:

- Passwords can be a maximum of 4 characters - consisting of 0-9, upper and lower case letters A-Z, or a combination of numbers and letters.
- If no password is entered, there is no restriction.
- Entering the wrong password or pressing ESC at the password prompt will return the keypad to the standard run-time display.
- Select EDIT > SETUP > MISC > PASWRD > CLEAR to delete all passwords.

NOTE: Subsequent attempts to RUN or EDIT a program do not require the password to be entered each time. You will be prompted to: **Use Last (F1)** or **Reset (F3).** Select **Use Last** to run or edit another program. Select **Reset** to require the next user to enter a password.

Chapter 6 - Programming Commands

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

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

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

DI

Distance Incremental ..... syntax - DI–r,–r [v1.00]

Command implemented in this firmware version

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

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

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

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

Note: Additional programming examples are found in the next chapter.

Summary of IDeal" Commands

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

IDeal™ Commands

Acceleration.....syntax - ACr [v1.00]

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

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

```
Examples: 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}
```

Break .....syntax - BR [v1.20]

Units: selected from the EDIT > SETUP > MECH menu
Default: n/a

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

Example:

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

Distance Absolute ..... syntax - DA–r [v1.00]

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

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

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

Examples: AC2 DE.5 VE12 DA3 GO {Moves to absolute position 3 units}
DA3 GO DA3GO {Moves *once* to absolute position 3 units}

DC

Distance to a Change.....syntax - DC–r [v1.00]

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

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, or change the state of one or more outputs.

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

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

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

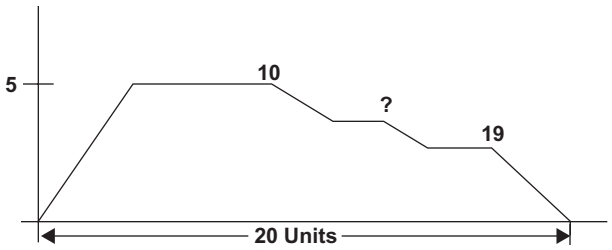
The following formula ensures the use of valid DC trigger positions:

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

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

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

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



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

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

AC.05 DE.09 VE30 DA6 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 *additional* 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 an *additional* 5 distance units, ramp down to 5 speed and continue until the final position is reached}

DE

Deceleration ..... syntax - DEr [v1.00]

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

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

Examples:	AC2 VE12 DA3 GO	{Sets acceleration and deceleration to 2}
	DE.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–r [v1.00]

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

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

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

EA

Enable/Disable Amplifier ..... syntax - EAi [v1.00]

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

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

Example: EA0 {Disables the amplifier on axis one}

EB

End of Block ..... syntax - EB [v1.00]

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

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

Examples:

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

EN

End of Routine..... syntax - EN [v1.00]

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

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

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

Example:

IF2,1 EN EB DI2 GO	{If input #2 is ON, stop the program, or returnto the calling program. If not, move 2 units}

FK

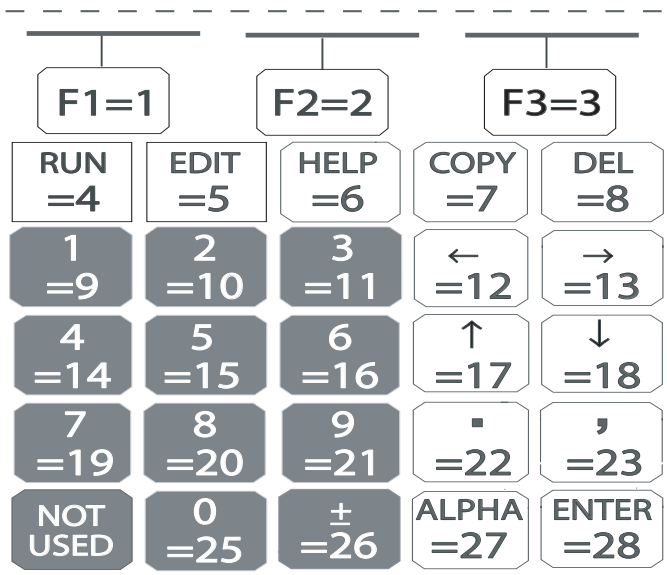
Function Key .....syntax - FKi,i,..,i [v1.00]

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

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

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

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



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

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

Example of a 3-screen menu program:

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

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

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

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

Go Home .....syntax - GH–r [v1.00]

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

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

The motor will move at the GH velocity (n) and direction (±) specified until it either finds a home limit switch or determines that it 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.

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

Examples:

AC.5 DE.5 GH-20 {Go Home in the negative direction at a speed of 20}

AC.5 DE.5 GH20 {Axis one Go Home in the positive direction at a speed of 20}

GI

Go Immediate ..... syntax -GI or GIi [v1.00]

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

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

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

VE1 DI20 GI MS1, Axis #1 is moving TD2

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

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

USING THE GI COMMAND - EXAMPLES
The following examples are provided to help further explain the use of the command:

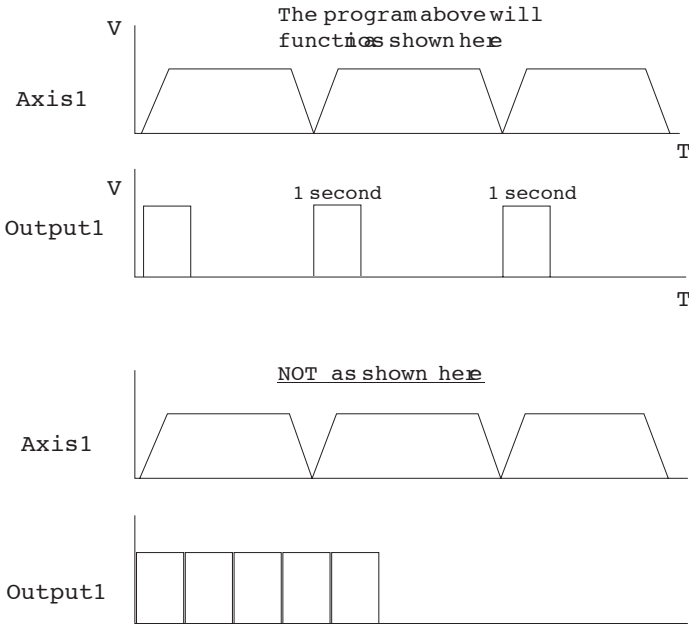
A. If a GI move is in progress and an additional move is commanded on the same axis, the additional move will not begin until the GI move has completed. For example:

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

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

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

LP VE2 DI10 GI OT1 TD1 OT0 EB



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

Go (Start a Move) .....syntax - GO or GOi [v1.00]

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

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

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

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

GS

Gosub.....syntax - GSi and GS[name] [v1.00]

Units: n/a
Range: i = 1-400, [name] = any legal program name
Default: n/a

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

Example: DI10 GS[Part A] GO {Run program "Part A", return and make a 10 unit incremental move}

GT

Go to Program..... syntax - GTi or GT[name] [v1.00]

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

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

Example: IF10 GT[PART A] EB {IF input 1 is on and input 2 is off, jump to program "Part A"}

IF01 GT20 GT30 EB EN {IF input 1 is off and input 2 is on, run program 20. Program 30 will never run. Use the GS command if you want to return to this program and *goto* program 30}

If ..... syntax - IF (See Below) [v1.00]

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

Syntax(s): IFi,xx...(checks input range beginning at input "**;**")
IFxx... (assumes first input is input 1)
IF (mathematical expression) or expressions (2)
Range: i = starting input number, 1-8
x = 0; input high
x = 1; input low (grounded)
x = *anything else* (ignore input changes)
expression = any valid expression (defined in math and variables section).

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

Note: An End of Block (EB) command must be used with every IF command.
IF blocks can be nested up to 16 levels deep.

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

Example:

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

IV

Input Variable ..... syntax - IVi,(variable),*min,max* [v1.00]

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

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

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

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

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

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

Example:

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

LP

Loop .....syntax - LPi [v1.00]

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

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

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

The *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.

Loop Until Condition True..... **syntax - LU (See Below)** [v1.00]

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

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

Syntax: LUi,xx…
LUxx…
LU(Mathematical expression) or expressions (2)
where,
i = starting input number, 1-8 (SmartStep), 1-16 (SmartDrive)
x = 0, (Input Off)
x = 1 (Input On)
x = *anything else* (ignore input changes)
Mathematical expression = Any valid conditional or logical expression

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

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

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

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

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

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

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

Loop While Condition True **..... syntax - LW (See Below)** [v1.00]

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

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

Syntax: LWi,xx…
LWxx…
LW(Mathematical expression) or expressions (2)
where,
i = starting input number, 1-8 (SmartStep), 1-16 (SmartDrive)
x = 0, (Input Off)
x = 1 (Input On)
x = *anything else* (ignore input changes)
Mathematical expression = Any valid conditional or logical expression

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

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

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

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

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

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

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

Move Continuous .....syntax - MC+ [v1.00]

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

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

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

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

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

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

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

If a motor is making a move when it comes to the end of a program, the motor will continue moving, even after the program ends. This allows you to:
- Put different MC moves in different programs and select different speeds by running different programs.
- Change speeds based on Binary or BCD program select lines

- Call MC moves as subroutines
- Run from “hosted” RS-232C mode, where the computer commands speed changes
- Run another program from the keypad that does not violate MC syntax. So you could run another program from the keypad to change speeds, move the other axis, manipulate I/O, interface with an operator or calculate arithmetic.

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

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

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

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

Example 3: Demonstrates 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}
```

MS

Message to Display..... syntax - see below [v1.00]

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

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

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

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

Examples:

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

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

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

ON Command (On Event) .....syntax - ONn,GTx, ONn,GSx, ONn,0 [v1.8]

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

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

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

n: L On EOT Limit

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

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

Example: Using ON to handle an EOT event

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

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

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

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

OT

Output ..... syntax - see below [v1.00]

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

Syntax: OTi,xx...(sets output states starting with output "**;**")
OTxx...(assumes first output is # 1)

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

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

Example:

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

" "

Quote..... syntax - *Any ASCII character* [v1.00]

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

The " " (Quote) command transmits a string out the serial communications interface. A "" without any string will transmit a carriage return character (ASCII 13). Variable values may be transmitted over a serial interface with the (TERM) variable (see page 7-3).

Example:

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

RG

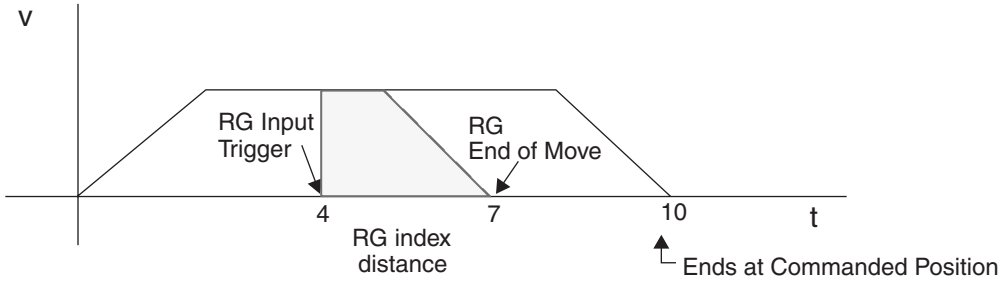
Registration .....syntax - RGr [v1.00]

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

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

VE2 AC.1 DA10 RG3 GO

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



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

Note: Registration Input is only configurable on input #1 for axis #1.

System Performance when Using the RG (Registration) Command
The input capture delay is 5µs. Worst case position error is 9 steps at 50 rps.

Set Position ..... syntax - SP–r [v1.00]

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

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

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

Square Root..... syntax - SQr,(var) [v1.00]

Units: n/a
Range: 0.0001 to 214748.3645
Default: n/a

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

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

Program: (SQRESULT)=0 SQ27,(SQRESULT)

The returned value in (SQRESULT) would be 5.28.

ST

Stop On Input ..... syntax - STn or ST#n [v1.00]

Units: n/a
Range: 0-8 (Inputs)
#1 (Axes)
Default: n/a

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

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

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

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

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

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

TD

Time Delay ..... syntax - TDr [v1.00]

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

Delay r seconds before executing the next command.

Example: VE50 DI4 GO OT11 TD.5 OT00
{Move 4 units, turn outputs 1 and 2 on, delay 0.5 seconds, and turn outputs 1 and 2 off.}

See also: System variable (TIME), in Chapter 7, Programming Your Application.

VE

Velocity ..... syntax - VEr [v1.00]

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

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

Example:

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

Wait..... syntax - see below [v1.00]

Units: n/a
Range: i = starting input number, 1-16
x = 0; input high
x = 1; input low (grounded)
x = *anything but 1 or 0*; ignore the input level
expression = any valid expression as defined in the math and variables section.
Default: n/a

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

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

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

Example:

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

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

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

Refer to Chapter 7, Programming Your Application, for more details and examples of how to use the above operators, functions, and expressions.

Chapter 7 - Programming Your Application

The purpose of this chapter is to give the programmer the information necessary to begin developing an application with a SmartStep. Also in this chapter are several practical examples that may be copied and modified. More program examples are available in *Application Developer* by selecting Help > Command Help.

SmartStep Programming Overview

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

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

The SmartStep 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 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 is 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 *Chapter 5, Programming with Serial Communication*, there is more information on serial commands, such as Setup, Immediate Status, and Supervisory Commands. Also included in Chapter 5 is a list of "Commands Not Available in Hosted Mode."

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

Variables and Arithmetic

Variables

The SmartStep will accept a variables in a commands instead of a constant. 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 SmartStep 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
(AI1) thru (AI6)	Analog Input 1 thru 6	Read Only
(AROWREL)	Current status of any of the four arrow keys	Read Only
(CPOS1)	Commanded position of axis 1	Read Only
(EPOS1)	Encoder position of axis 1	Read Only
(VEL1)	Commanded velocity of axis 1	Read Only
(POS1), (POS2)	Current Position of axis 1-2	Read Only
(#F1) thru (#F50)	Non-volatile, limited use, user system variables	Read/Ltd. 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 of Built-In Variables:

• (PIECES)=10	Assigns 10 to variable
• (SPEED)=(AI4)*(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)=(AI6)	Reads in temperature from an analog input
• (#F1)=(PIECES)	Stores the value of Pieces in the FLASH variable #F1

Using the Built-In Variable (AROWREL)

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

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

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

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

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

[RIGHTARROW]{Program #13}
MC+                     {Enable MC mode}
AC.1                    {Start MC move}
VE-1                    {Move in negative direction}
GO
```

```
LP
  IF(AROWREL)=1      {Check status of arrow key}
    VE0              {Stop MC move on key release}
    GO
    GT1              {Return to main program}
  EB
EB                   {End loop block}
```

Non-Volatile Variables

The non-volatile variables (#F1)-(#F50) are fifty 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 FLASH.

CAUTION

Caution must be used when using these variables. Since FLASH have a limited read/write lifetime (100,000 writes before failure), variable values that change frequently should not be stored as FLASH system variables. Examples include loop count variables, and POS1 and POS 2 variables. The SmartStep will allow only 1,000 FLASH writes between power cycles. This limit has been set to prevent a simple programming mistake or misunderstanding from permanently damaging the SmartStep'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 FLASH 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 FLASH variables, but does not generate any *writes* to the FLASH, so the lifetime of the FLASH 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)
(#F1)=(LENGTH)            Loads the length into non-volatile system variable (#F1)
(#F2)=(SPEED)             Loads the speed into non-volatile system variable (#F2)
EN
```

[PARTS]	[PARTS] runs on power up, unless new parameters are entered.
(LENGTH)=(#F1)	Load the part specific variable from the non-volatile variables.
(SPEED)=(#F2)	
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 SmartStep 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 SmartStep fixed-point variable storage only supports 4 places to the right of the decimal place (32 bit storage of fractional decimal number).

Instead enter:

• (X)=1		
(X)=(X)+2	or	(X)+=2
(X)=(X)-3	or	(X)-=3

and

• (Length)=(Total)*3.125	Multiply by the significant figures.
• (Length)=(Length)*.01	Then move the decimal place.
• **or** (Length)=(Total)/32	The SmartStep 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.

Logical Operations on Expressions (&&,||)

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

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

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

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

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

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

Expressions

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

Examples:

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

Other Typical Programming Examples

The following example programs will give you an idea of how the IDeal command language can be used to solve simple tasks. More extensive and elaborate example programs can be found in the DEMOS.idc file that came with your IDC CD-rom. This file can be accessed from application developer.

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

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

To create a Message and input a Variable

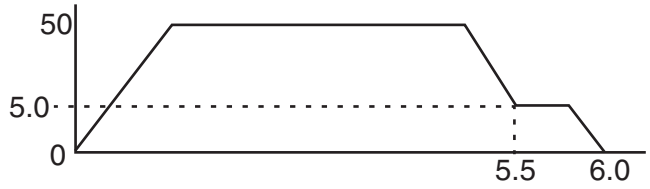
[GET PARTS]	Name the subroutine
MS1,””	Clears the Display
MS1,”How many?: ”	Writes string beginning at character 1, top line
IV12,(PIECES)	Waits at 12th character for the # of pieces.
MS1,””	Clears the Display
MS1,”How long?: ”	Writes string beginning at character 1, top line
IV12,(LENGTH)	Waits at 12th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

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

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

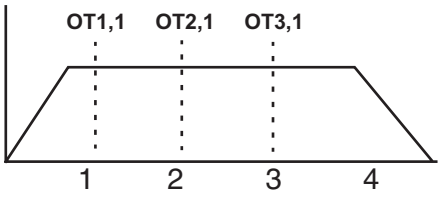
Fast In, Slow Feed Move (Using the Distance to Change (**DC**) command)

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



Turning On an Output on-the-fly

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



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

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

Reading an Analog Input

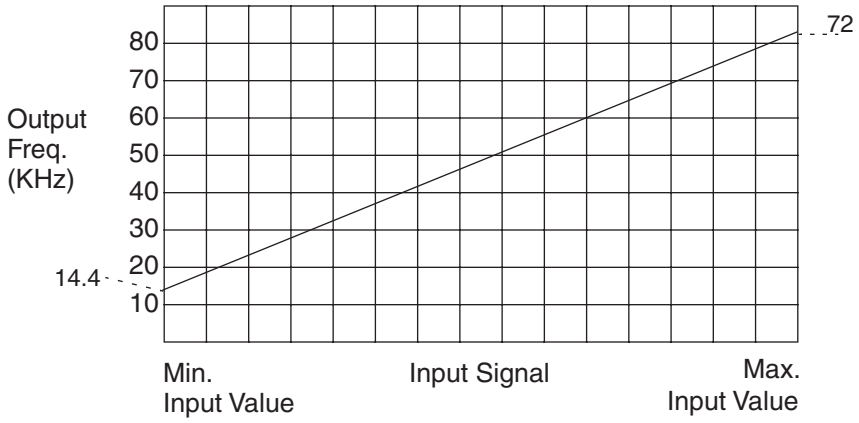
The value of the analog system variables (AI1-AI6) are scaled from 14,400 to 72,000 Hz. This value is actually a scaled frequency read from the OPTO module representing the analog signal. These input values are updated every 16 milliseconds . If your program needs to display this value in units such as VOLTS, you will need to scale the value to VOLTS in your program. The scaling factor depends upon the type of OPTO module used. For example: a "J" thermocouple uses a different factor than a "K" thermocouple. Due to slight variances in the output frequency from module to module, it is recommended that the OPTO be calibrated by querying the corresponding AIx value with no input signal connected to the OPTO. This value should be used as the zero input reference frequency.

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

(VOLT)=(AI2)	Read the value of analog input #2 into variable volts
(VOLT)=(VOLT)-14400	Remove frequency offset
(VOLT)=(VOLT)*1.736	Scaling factor multiplied by 10,000
(VOLT=(VOLT)*.0001	Scaling back to volts

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

(TEMP)=(AI5)	Read in temperature from analog input
WT(AI3)<45000 GO	Wait for analog input 3 <45000 (<5.3 VDC using the previous example) before moving
IF(AI2)<45000 GO EB	Go if analog input 2 < 45000



Chapter 8 - Programming with Serial Communication

Overview

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

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

This chapter is divided into 4 sections. Section 1 covers the installation of *Application Developer*. Section 2 covers *Using Application Developer* to setup and program SmartStep systems. Section 3 covers common RS-232C details, including baud rate settings as well as hardware and daisy-chaining information. Section 4 provides details on all of the RS-232C commands that *Application Developer* employs. This section will be useful to users who are not using Windows, or who plan to run the SmartStep 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.

This page intentionally left blank.

Section 1: Application Developer Software

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

Installing Application Developer in Windows 95, 98, or NT

1. Insert the *Application Developer* CD in your CD-ROM drive.
2. Click on Start.
3. Click on Run.
4. Type the following in the Command Line box that appears (replace "d" with your CD drive letter if different): **d:\App_Dev\Setupex.exe**
5. Click OK.
6. Follow the installation instructions on the screen.
7. Restart Windows (required).

If You Need Help Installing Application Developer

There are no known installation problems with Application Developer. Please call the IDC Electric Applications Department at (800) 544-8466 or (704) 588-5693 (from outside U.S.) if you need assistance installing this software.

Section 2: Using Application Developer

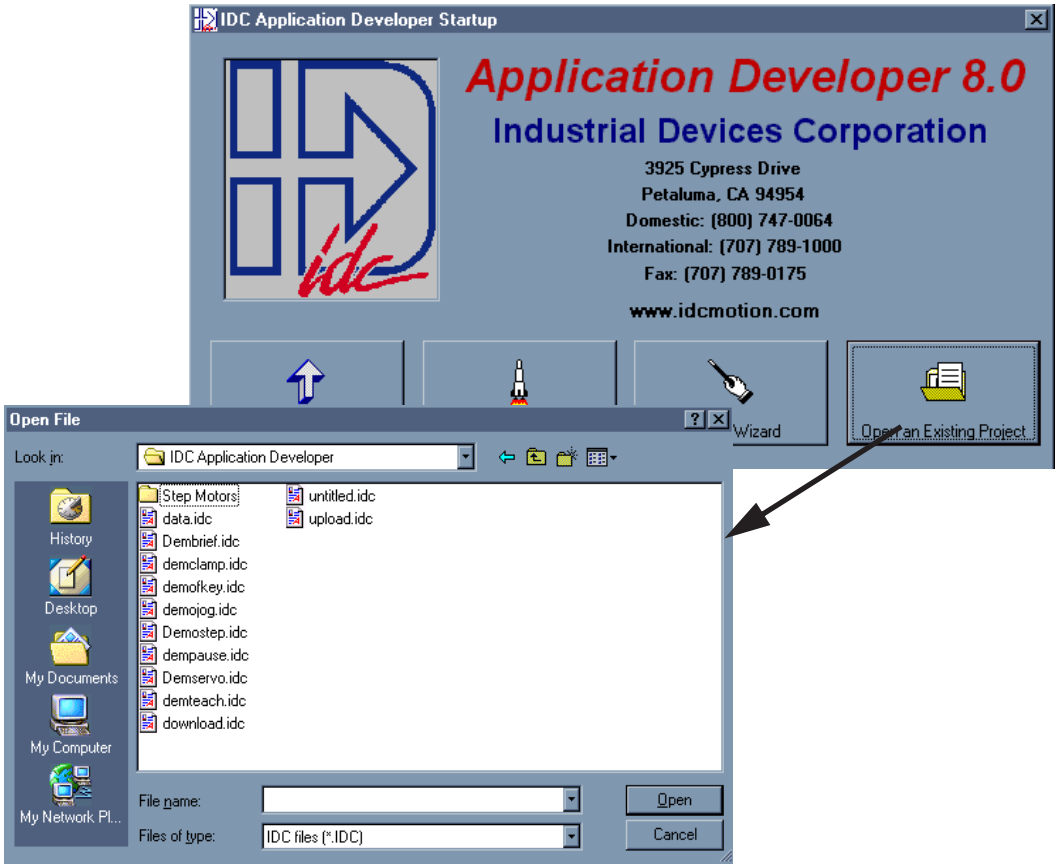
Application Developer's graphical environment helps you setup and program your SmartStep from your computer. It guides you through configuring your control, following the same steps and menus as the keypad configuration. Complete details on setup menus and choices can be found in Chapter 5 - *Configuring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

Using the Setup Wizard

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

Open an Existing File

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

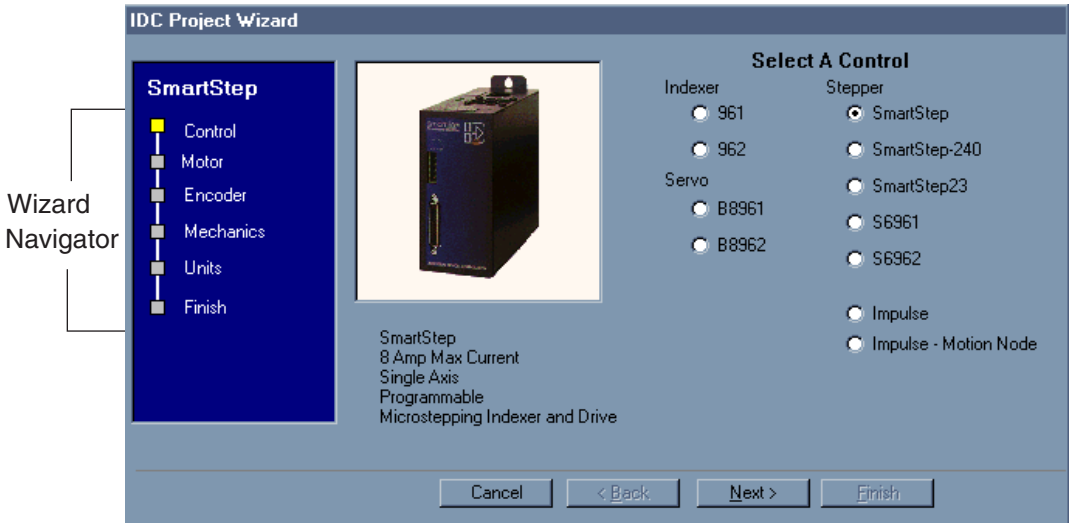


New Project - Using the Setup Wizard

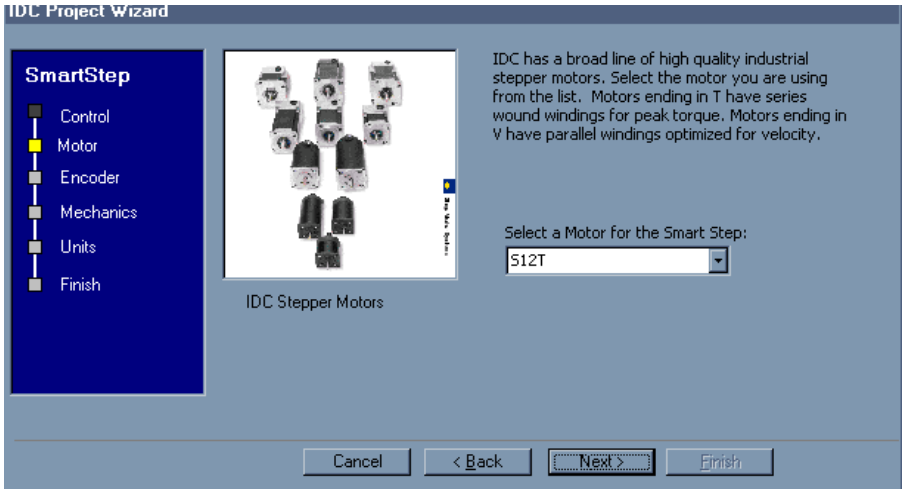
1. Click on the **New Project** button to start the Setup Wizard.

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

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

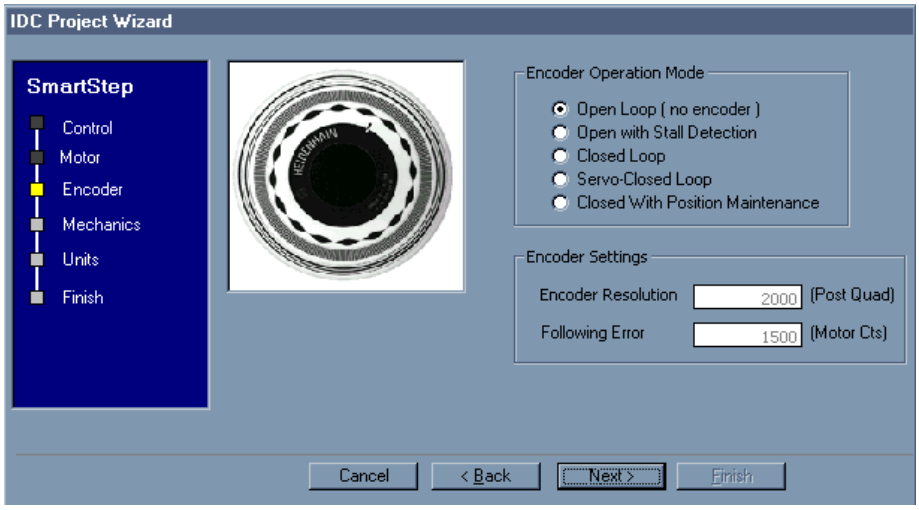


3. Click on the Next button and the Motor setup window will appear.



4. From the pulldown menu select the IDC motor you will be using and the Wizard will calculate the rest. Select Other if you are using a non-IDC motor.

5. Click on the Next button and the Encoder setup window will appear.



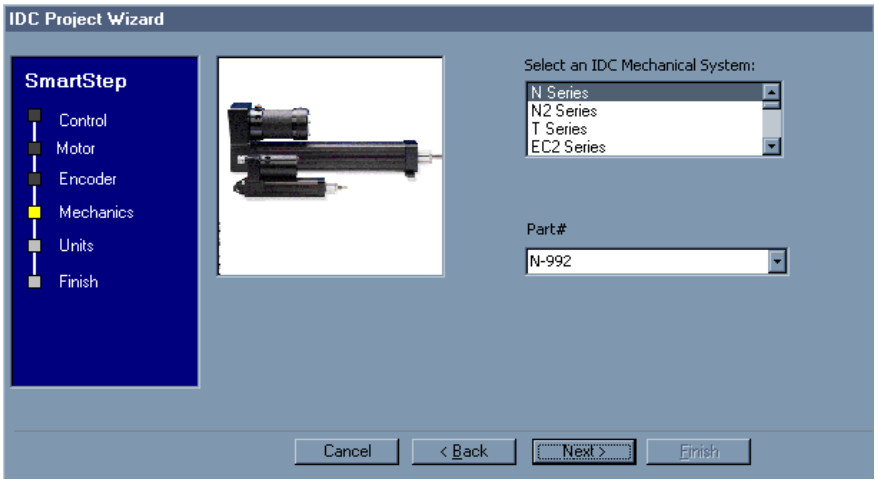
6. If you will be connecting an encoder, select one of the Encoder Operation Modes. Encoder modes are described in Chapter 5.

7. If you have made a selection other than Open Loop (no encoder), you may wish to edit the Encoder Resolution and Following Error parameters at this time.

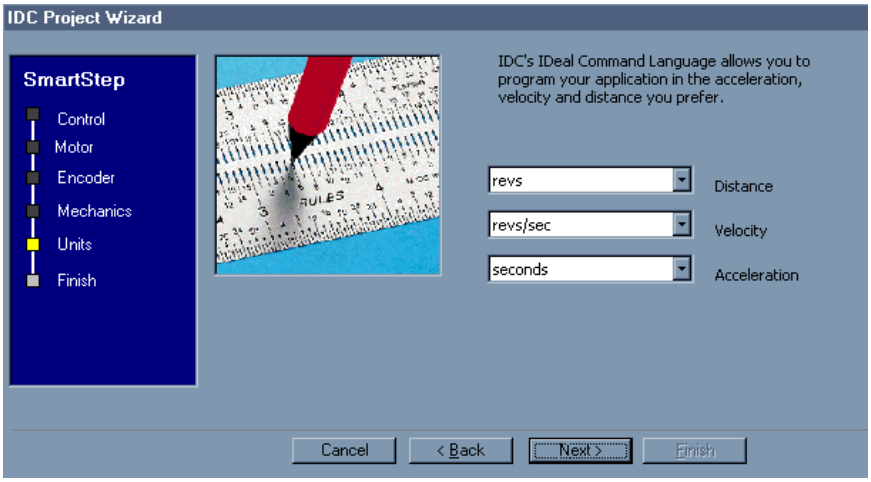
8. Click **Next** and you will be in the Mechanics setup window. Mechanical System setup is particulary easy if you are using IDC systems.

   Select the IDC Mechanical System and Part # you intend to control and the Wizard will calculate the rest. Select Other if you are using a system not otherwise specified.

Thomson Actuator	Select IDC equivalent
TN	N2
TC2	EC2
TC3	EC3
TC4	EC4
TC5	EC5

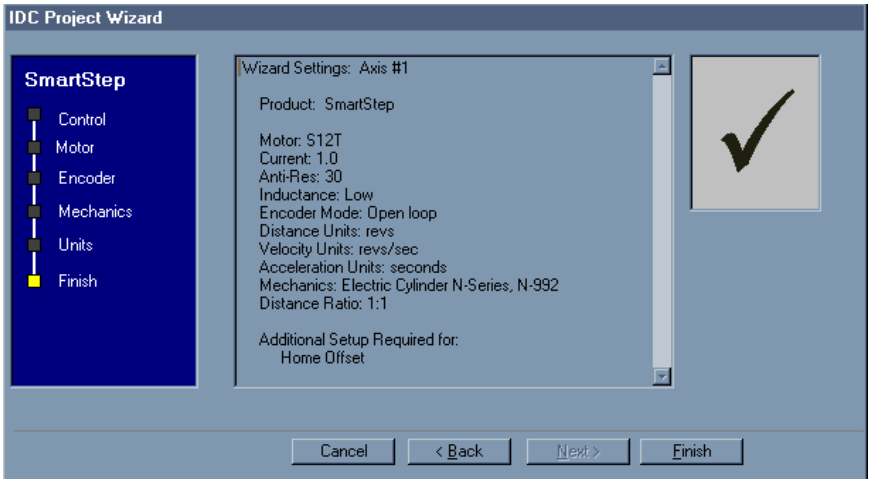


9. Clicking on the **Next** button will bring up the **Units** setup window. The SmartStep lets you program the control in the units that work best for your application.



10. Select your preferences from the pulldown menus and click on **Next**.

11. When all parameters have been configured, the following window will appear. This display gives you the opportunity to review the settings you have made and return (using the Back button) to any parameter you may wish to change at this time.



12. When you are satisfied with the setup of your system, click on **Finish**. Note that the Wizard Navigator now indicates that all axes and parameters have been configured.

After Using the Setup Wizard - More Configuration Parameters

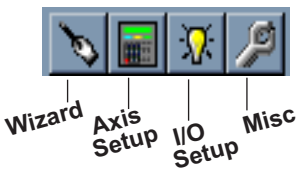
Your basic system setup parameters have been configured by the Setup Wizard, and though the system is functional, your application will probably require further configuration and fine-tuning.

The remainder of this chapter provides the same setup information as configured by the Setup Wizard plus more detailed information on each parameter to allow more control of your system and greater capacity for applications.

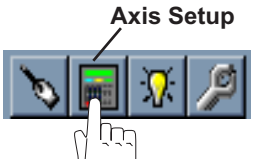
The Setup Wizard remains available at any time by clicking on the **Wizard** button. The **Axis Setup**, **I/O Setup**, and **Misc** toolbar buttons provide access to the setup parameters that will be explained throughout the remainder of this chapter.

Setup Tool Bar Buttons

To access the setup windows and menus, simply click on the desired toolbar buttons as shown here:

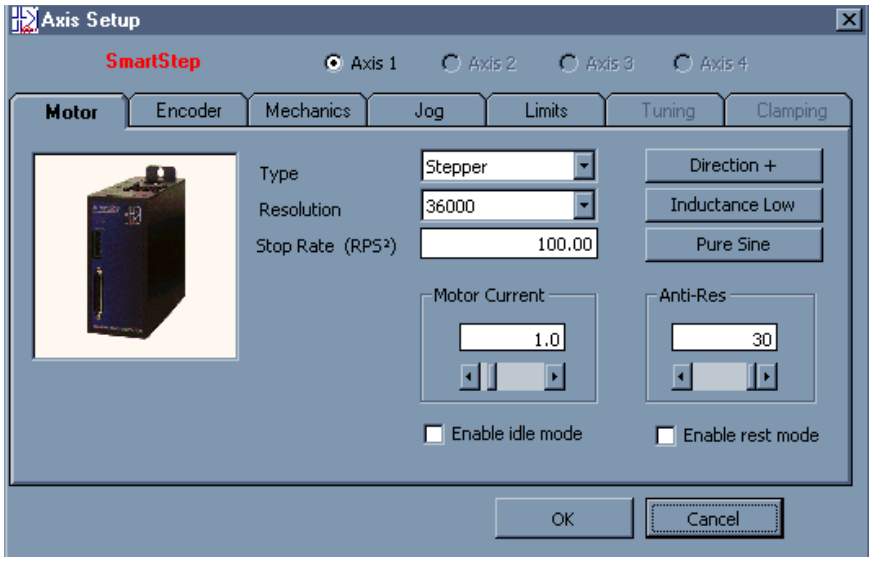


Axis Setup



Click on the Axis Setup button.

The first menu to appear will be the Motor menu, as shown below:
Note: please refer to Chapter 5 for detailed descriptions of setup parameters.



Axis 1

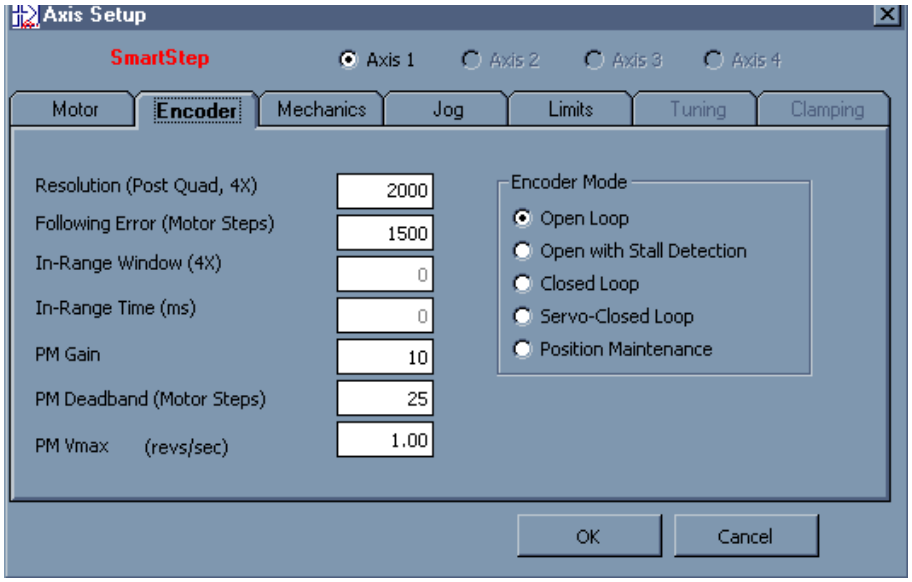
Axis 1 is automatically selected (SmartStep is a single-axis drive). Configure each parameter as it applies to your application (Motor, Encoder, Mechanics, Jog, Limits).

Motor Menu (shown above)

Settings for Drive Type, Resolution, Stop Decel Rate, and motor Directions are selected in the **Motor** menu. Motor **Type** automatically defaults to Stepper (no options).

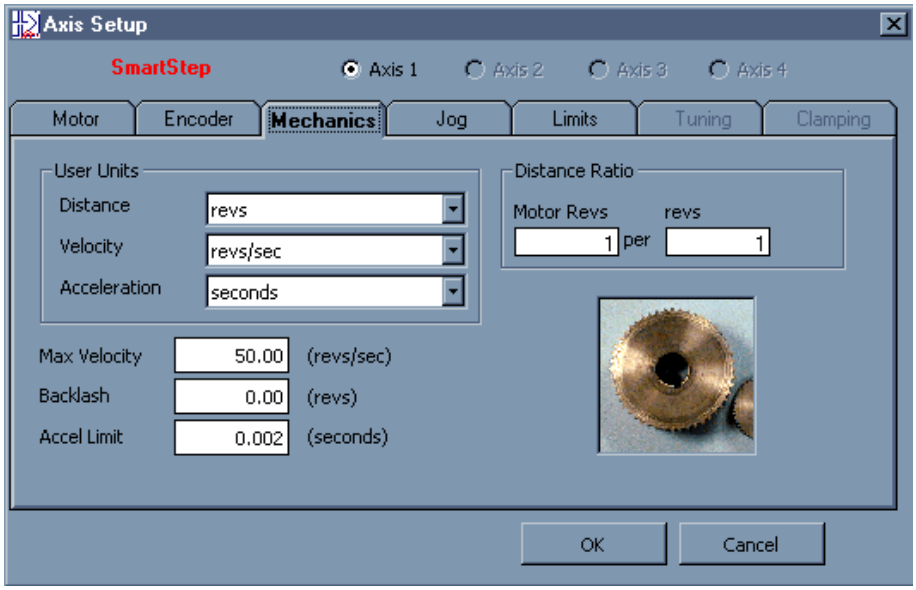
Encoder Menu

If you are not using an encoder, only the Encoder Mode must be configured. **Ensure that OPEN LOOP is selected if you are not using an encoder,** and skip to the Mechanics menu.



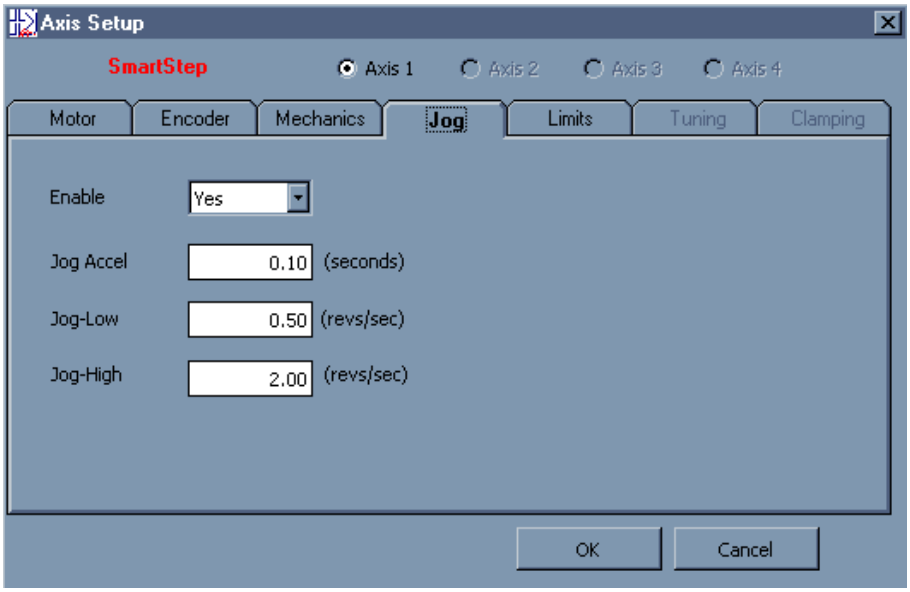
Mechanics Menu

The Mechanics menu allows you to program distance, velocity, and acceleration units convenient for your application. This menu also allows you to set a maximum allowable speed and acceleration for each axis.



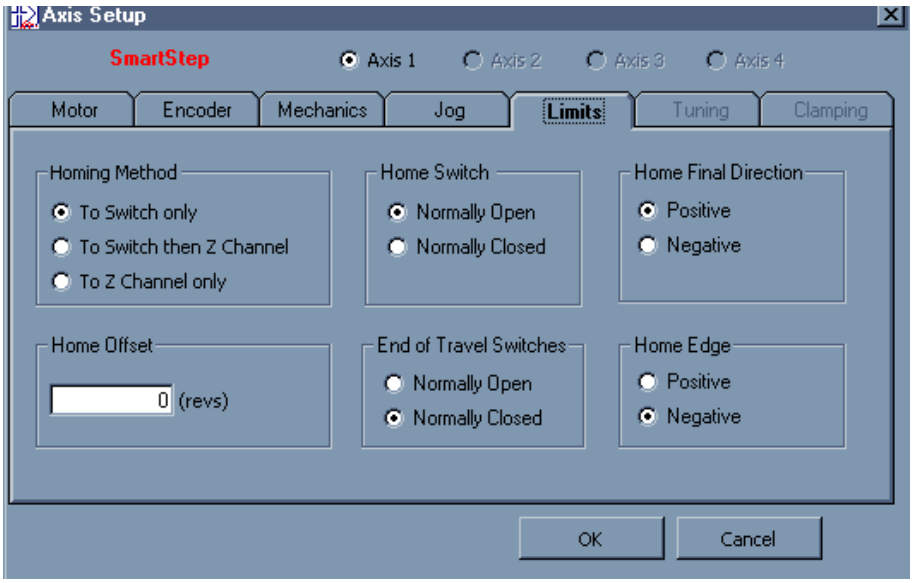
Jog Menu

The parameters which control your jog operation are configured using the Jog menu shown below.

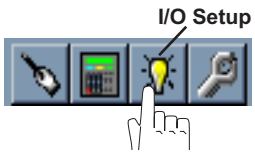


Limits Menu

Your SmartStep 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.

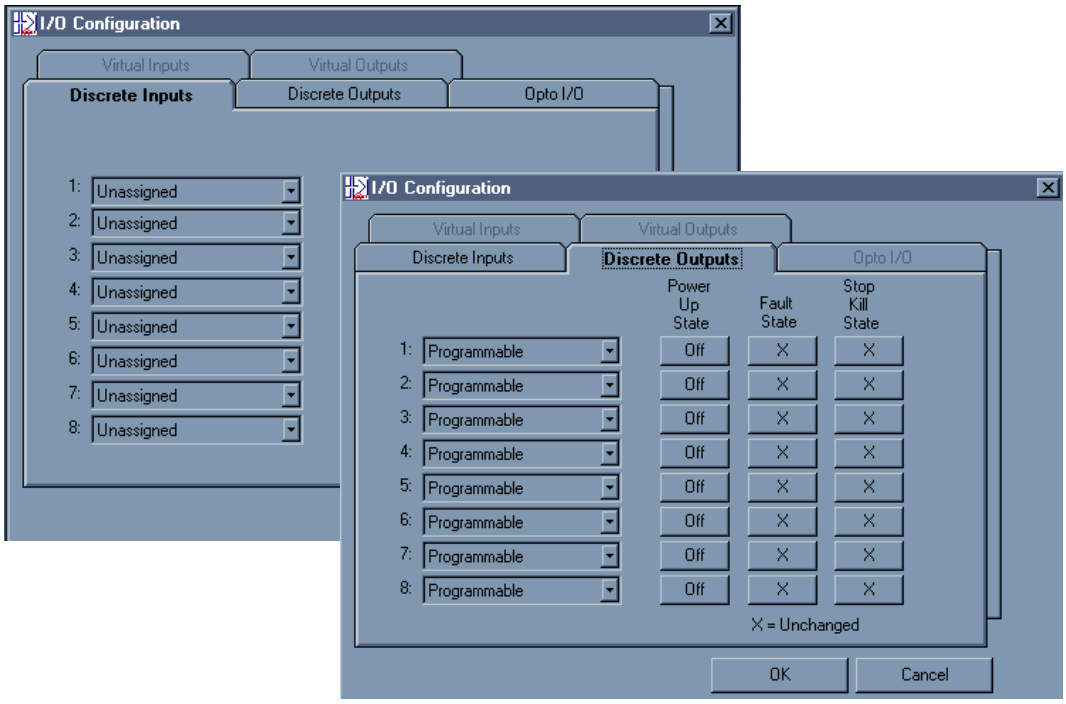


I/O Setup



Click on the I/O Setup button.

To define a dedicated function for each discrete input and output, scroll through the pulldown lists and select from the available choices.



Miscellaneous (Misc) Setup

Click on the Misc (Miscellaneous) button.

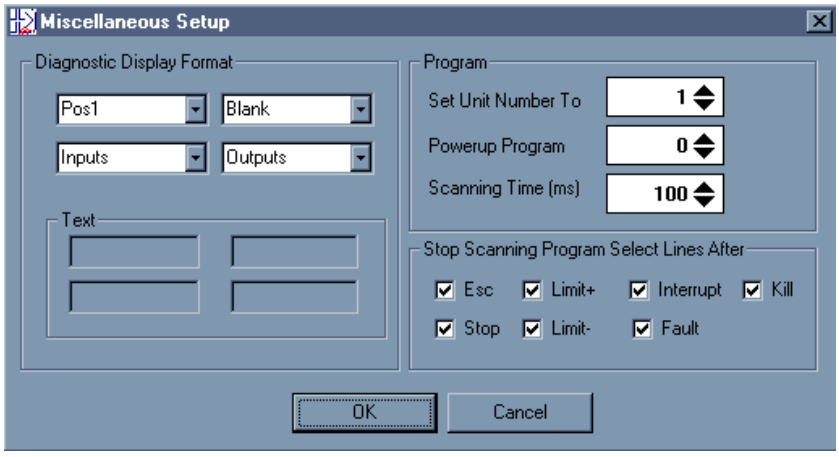
Misc Setup contains two configuration categories that include options available in the PROGRAM SETUP and RS-232C SETUP menus detailed in Chapter 5.

Diagnostic Display Format

Diagnostic Display Format allows you to customize the data display on the IDC FP220 Keypad. This parameter does not apply if you do not have the keypad.

Program

a. Sets the SmartStep address (Set Unit Number To).
b. Sets the program to run on power-up (Powerup Program). No program will run if set to 0.
c. Sets the debounce (Scanning Time) of the program select inputs in milliseconds.

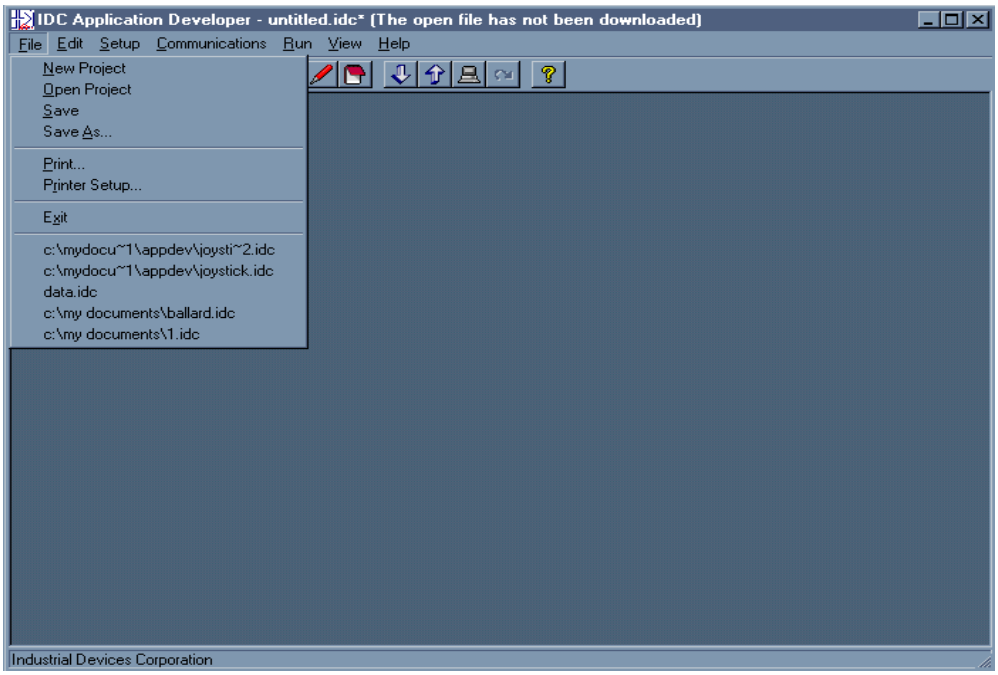


Stop Scanning Program Select Lines After...

Allows you to select the conditions under which program scanning stops.

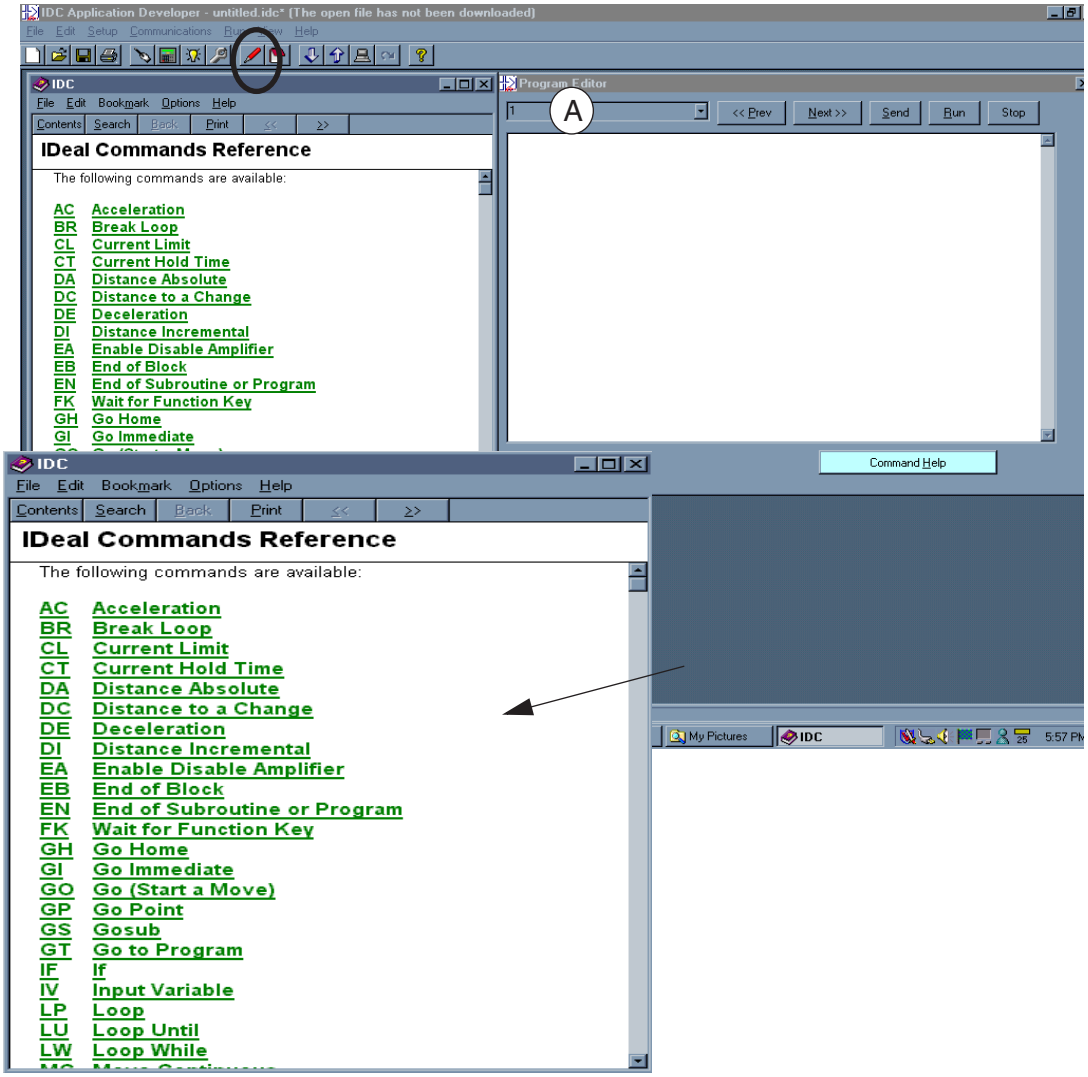
File Menu

Applications (programs and configuration files) may be stored on disk as DOS files. The default suffix is *.idc. The other selections under File are generic to all “Windows” applications.



Program Editor

The **Program Editor** features standard windows editing features. Cut, Copy, Paste, Undo, Delete, and Select All are available by pressing the right mouse button. An online HELP file may be reached by clicking on the Command Help button shown below.

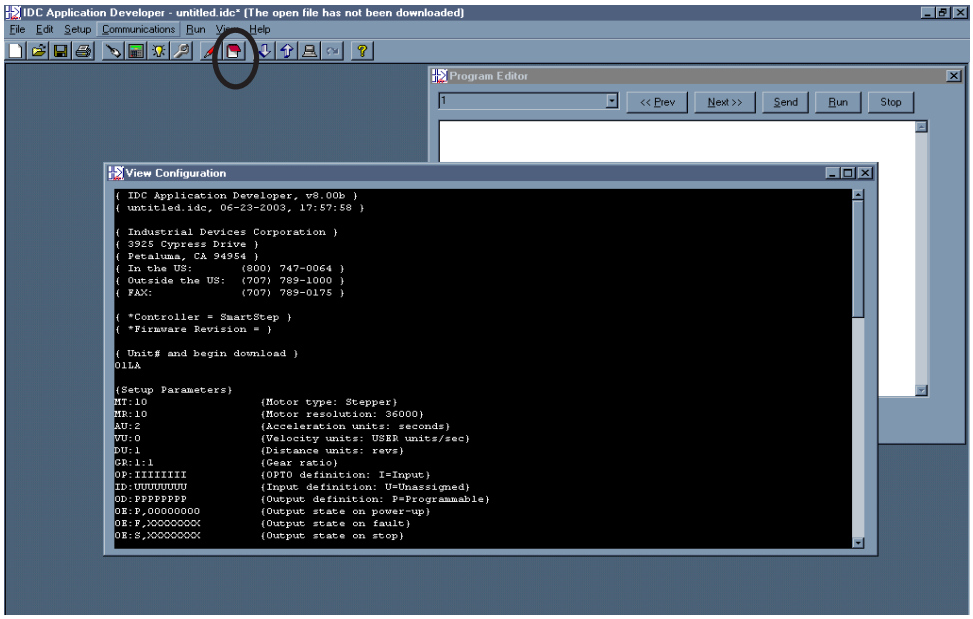


The drop down menu box (A) in the upper left hand corner shows the number and name of the currently active program, plus a list of up to 399 more programs. When the entire file is downloaded to the SmartStep, these program numbers correspond to the program numbers the controller uses for binary and BCD program selections.

Program comments are placed between brackets {comments}. These comments are not downloaded to the SmartStep Total program length, not including comments, is limited to 1k. Total program length with comments is 8k.

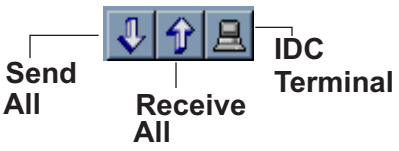
View Configuration

Click on the View Configuration button at any time to see your system configuration status. All configuration parameters are listed and may be viewed by scrolling the list.



Communications

All Communications functions may be accessed from the main menu bar. Send All, Receive All and IDC Terminal are selectable on the Toolbar as shown here:



Send All

Use *Send All* to download the application you have developed. In addition to motion programs, your application file will include the setup commands derived from the choices you made in the Setup dialog boxes. *Send All* completely configures the SmartStep 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 SmartStep. IDC recommends saving the commented version of your application before downloading.

Retrieve All

Use *Retrieve All* to upload the entire contents of a SmartStep control to a new file that can then be edited, downloaded to another SmartStep, or saved to a PC file for documentation purposes. This file contains the complete contents of the SmartStep 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 SmartStep.

Change Unit Number

Change Unit Number is used to set the device address of the control that *Send/Receive Program* uploads and downloads to on a single RS-232C daisy chain. Each unit must have its own unique software address. The Unit number of each control should be set *BEFORE* the units are connected in a daisy chain (the default address is one). *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 SmartStep control. It is very useful for troubleshooting interactive host/control communications.

Run Menu

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

Run is only accessible from the main menu bar.

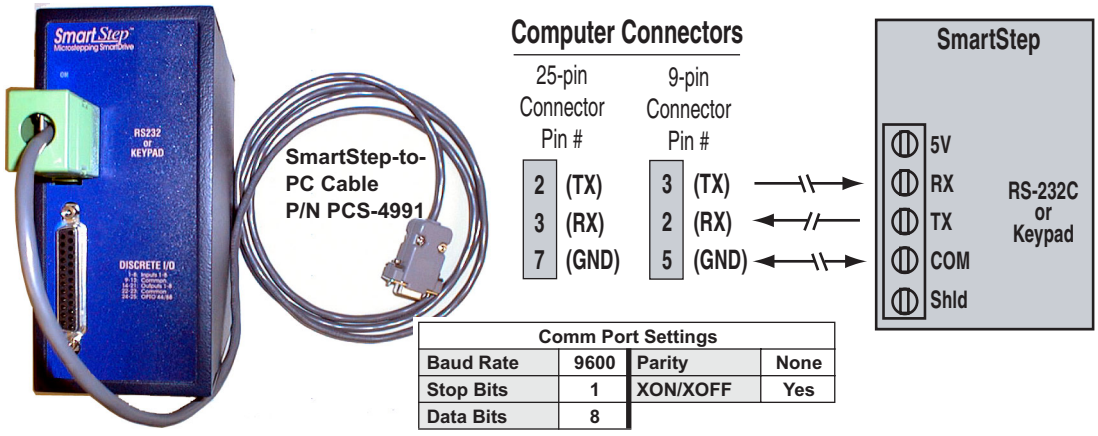
This page intentionally left blank.

Section 3: RS-232C Protocol

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

Making RS-232C Connections to the SmartStep

Make your RS-232C connection as shown below. The SmartStep-to-PC Cable (IDC P/N PCS-4991) shown below, is an ideal choice for making trouble-free connections.



Comm Port Settings			
Baud Rate	9600	Parity	None
Stop Bits	1	XON/XOFF	Yes
Data Bits	8		

Troubleshooting Serial Communication Problems

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

1. Test your terminal or terminal emulation software. Unhook the drive and transmit a character. An echoed character should not be received. If an echoed character is received, you are in half duplex mode. Jumper your host's TX and RX connections, and transmit another character. If you do not receive the echoed character, consult the manufacturer of the host's serial interface for proper pin-outs.
2. Host transmit (TX) must be connected to receive (RX) of the drive unit, and receive (RX) of the host must be connected to transmit (TX) of the drive. If communication fails, try switching connections on either the host or the drive.
3. Many serial ports require handshaking. Jumper RTS to CTS, and DSR to DTR (see table).
4. Configure the host to the identical baud rate, number of data bits, number of stop bits, and parity.
5. Receiving double characters (XX) when entering single characters (X), indicates your computer is set to the half duplex mode. Change to the full duplex mode.
6. Check your grounds. Use DC common or signal ground as your reference. Do not use earth ground or shield.
7. Check your cable length. If any cable is over 50 ft. long, you should be using a line driver, optical coupler, or shield. Shields must be connected to earth ground at one end only.

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

Daisy Chaining SmartSteps

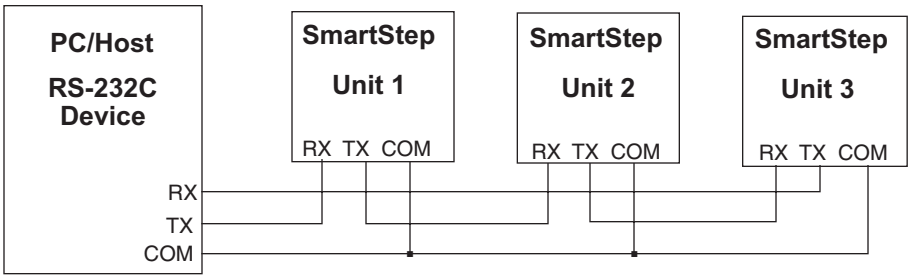
Your SmartStep also supports daisy chaining. The unit address (range 1-99) can be set via the keypad, through Application Developer, or with a terminal program using the Unit Number (UN) command, or the entire chain may be addressed at once using the Auto-Address (AA) command.

Rules for Daisy Chain Operation

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

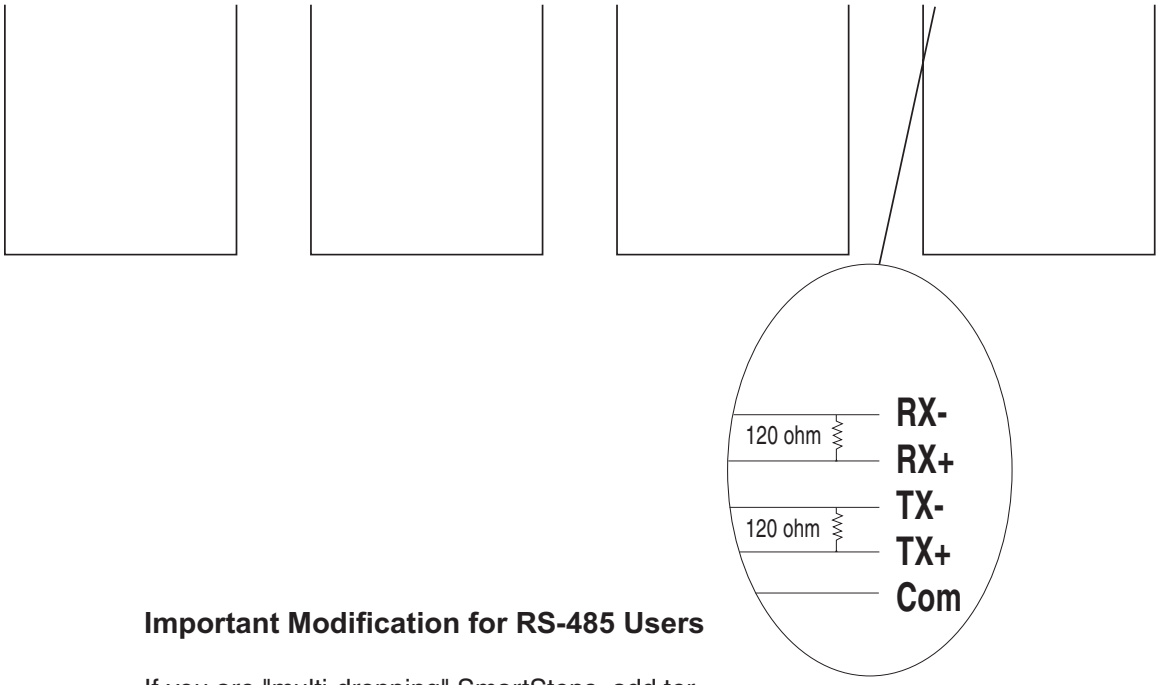
Please call IDC if you need to daisy chain more than 99 drives.

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



RS-485 Protocol

SmartStep Multi-Dropping



Important Modification for RS-485 Users

If you are "multi-dropping" SmartSteps, add termination resistors (120 ohms each) across RX and TX connections of the *last unit* as shown to the right.

Note: On early RS485 SmartSteps, **Com** may have been labeled as **Shld**. Functionally, they are identical.

This page intentionally left blank.

Section 4: RS-232C IDeal™ Command Reference

Overview

Though we strongly recommend taking advantage of the capabilities and convenience of *Application Developer*, you can configure, program, and run a SmartStep from any RS-232C terminal or computer. If you do not intend to use the Windows software tools we provide, you will need to use the IDeal™ RS-232C command listings that follow.

"Hosted" or "interactive" motion control from a PLC or PC is also a common mode of operation. You can write your control programs in your language of choice (BASIC, C, ladder, etc.).

RS-232C SmartStep operation is divided into four categories of commands. The first category is ***Serial Setup Commands***. These are the commands that IDC's *Application Developer* program uses to configure the SmartStep according to the choices made in the SETUP dialog boxes. These ***Serial Setup Commands*** include the syntax of the command, but the full command definitions and examples are found in *Chapter 5 - Configuring Your System*.

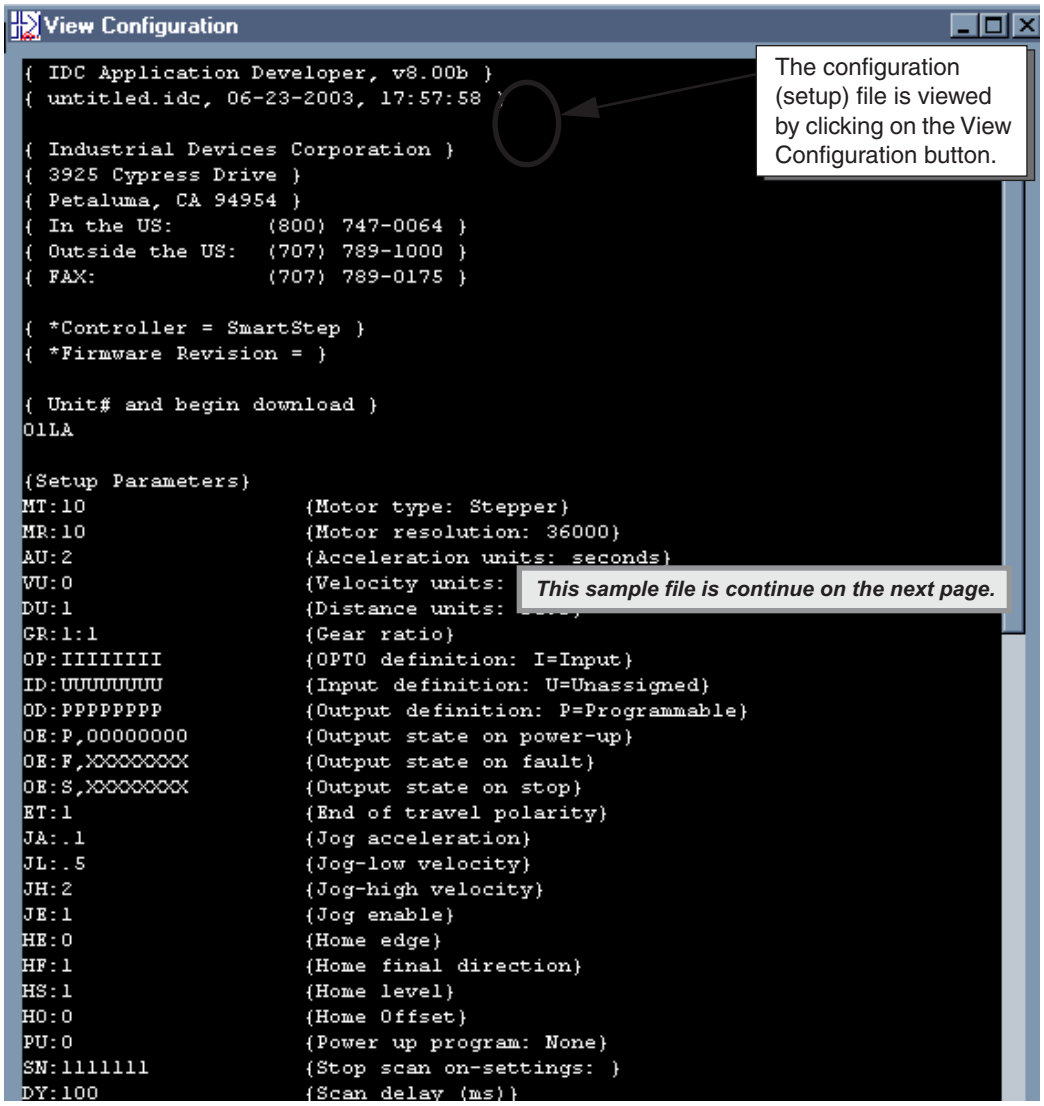
The second category is ***Serial Programming Commands***. These are commands that can be executed over RS-232C or downloaded to the SmartStep's non-volatile memory for later execution. This category of commands is identical in syntax and functionality to the equivalent keypad command. These run-time RS-232C commands are listed in the RS-232C Command Reference, but the full definition and examples are listed in *Chapter 6 - Programming Commands*.

The third category of Serial commands is ***Serial Immediate Status Commands***. These commands bypass the normal command buffer and are executed immediately, regardless of what else the SmartStep has been asked to do. These commands include instantaneous position reporting, drive status, and emergency kill and stop commands.

The final category is ***Serial Supervisory Commands***. These are the actual uploading and downloading of the setup and program parameters. Once an application with setup parameters and command has been created, these commands are used to download and upload the file to and from the SmartStep.

Sample *.idc File

To familiarize yourself with IDeal™ RS-232C programming commands, review the following example of a typical file created in *Application Developer* for download to a SmartStep. You will need to generate a similar file to configure your SmartStep. Individual programs can be downloaded in such a configuration file, or downloaded separately at run-time. the SmartStep can also execute run-time commands in an "immediate" mode outside of any program. See the *Command Syntax* section of this chapter for more details.



Sample configuration file continued:

```
EM:0                    {Encoder mode: Open loop}
ER:2000                 {Encoder resolution}
FE:1500                 {Following error}
IR:25                   {In Range}
IT:-10                  {In Range Time}
HM:0                    {Homing Method}
PG:10                   {PM Gain}
PV:1                    {PM Vmax}
BK:0                    {Backlash}
DF:1,0,4,5              {Display format: }
SR:100                  {Stop deceleration rate}
MD:0                    {Motor direction}
MV:50                   {Maximum velocity}
AM:.002                 {Accel limit}
AH:0                    {Anti hunt}
AW:0                    {Anti-inertial windup}
FA:0                    {Accel feed-forward gain}
FV:0                    {Velocity feed-forward gain}
KI:0                    {Integral gain factor}
KP:0                    {Position gain factor}
KV:0                    {Velocity gain factor}
EL:0                    {Motor enable polarity}
FL:1                    {Motor fault polarity}
BV:5                    {Clamp break velocity}
CU:0                    {Clamp current units}
AR:30                   {Anti-Res}
MH:L                    {Motor inductance}
MI:1.0                  {Motor current}
WA:0                    {Waveform}
RE:0                    {Rest 0:disabled 1:enabled}
IL:0                    {Idle 0:disabled 1:enabled}
WC:6                    {}
WG:0                    {}
WM: 0                   {}
WP:0                    {}

{Programs}

{End download}
01EX
```

Command Syntax

All *IDeal*™ commands use two letter UPPER CASE ASCII characters. Command delimiters can be a carriage return (<cr>) or space (<sp>) character.

The commands that follow define IDC's command syntax. A brief command description is given here, but the full command definition is found in Chapter 5. This listing is intended only to help programmers with command syntax. *IDeal* programming commands are defined in Chapter 6.

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

Letter or Symbol	Description
<n>	Unit address number is optional with RS-232C, and the command is sent to all units if no address is specified. All Status commands require an address. *Example:* <n>**AUi** RS-485 users must address all commands.
,	Field separator.
a	Alphabetic character.
h	Hexadecimal number.
i	Decimal integer number.
r	Decimal real number (up to 4 places to the right of the decimal).
:	A colon (:) is a neutral character. It can be used in a command to make it more readable to the programmer. For example OP:OOOOIIII is easier to understand than OPOOOOIIII. **Note**: The colon is required in GR command and is not neutral. (Unit Ratio). *Example:* **GR4:1**

Some *IDeal*™ commands request a response from the control. Responses will always be preceded by an asterisk (*) which notifies the other controls on a daisy chain to ignore the subsequent response characters preceding the next command delimiter. For example, the Input State (IS) command might return *AF09<cr> Your computer program will need to mask the asterisk before decoding the value returned.

You can document your programs by placing comments between brackets. For example: {this is a comment}. To maximize program storage space, the control "strips off" these comments when a program file is downloaded.

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartStep according to the choices made in the SETUP dialog boxes.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>AM</td><td>Acceleration Maximum</td><td><n>AMr</td></tr>
<tr><td>AR</td><td>Anti-Resonance</td><td><n>ARi</td></tr>
<tr><td>AU</td><td>Acceleration Units
i=0 units/s^2 (where “units” is a string defined by the DU command)
1 rps^2
2 seconds
Example: AU0 (units/s^2 on axis one)</td><td><n>AUi</td></tr>
<tr><td>DF</td><td>Display Format
Configures the four keypad run time display quadrants. DF takes four parameters where i is an integer representing a display data type per quadrant. User defined text is limited to 10 characters per field.

<table>
<tr><td>i = 0</td><td>Blank</td><td>i = 8</td><td>*CUR2</td></tr>
<tr><td>i = 1</td><td>POS1</td><td>i = 9</td><td>INPUTS</td></tr>
<tr><td>i = 2</td><td>POS2</td><td>i = 10</td><td>OUTPUTS</td></tr>
<tr><td>i = 3</td><td>POS1 + UNIT</td><td>i = 11</td><td>OPTOS</td></tr>
<tr><td>i = 4</td><td>POS2 + UNIT</td><td>i = 12</td><td>SA_STATUS1</td></tr>
<tr><td>i = 5</td><td>VEL1</td><td>i = 13</td><td>SA_STATUS2</td></tr>
<tr><td>i = 6</td><td>VEL2</td><td>i = 14</td><td>SS_STATUS</td></tr>
<tr><td>i = 7</td><td>*CUR1</td><td>" "</td><td>User defined text in quotes</td></tr>
</table>
*Not available</td><td><n>DFi,i,i,i
<n>DF”Text”,i,i,i</td></tr>
<tr><td>DU</td><td>Distance Unit Label
i = 0 steps (fixes GR @ 1:1)
<table>
<tr><td>1=rev</td><td>5=cm</td><td>9=deg</td><td>13=arcmin</td></tr>
<tr><td>2=inch</td><td>6=mm</td><td>10=radian</td><td>14=%</td></tr>
<tr><td>3= mil</td><td>7=yard</td><td>11=grad</td><td>15=index</td></tr>
<tr><td>4=meter</td><td>8=foot</td><td>12=arcsec</td><td>16=µm</td></tr>
</table></td><td><n>DUi</td></tr>
<tr><td>DY</td><td>Scan Delay
Where i is the number of milliseconds
Example: DY500 (500 ms) Default is 100ms</td><td><n>DYi</td></tr>
<tr><td>EL</td><td>Enable Line Polarity
Fixed in SmartStep</td><td><n>EL1</td></tr>
<tr><td>EM</td><td>Encoder Mode
0 = Open Loop,
1 = Open Loop with stall detect
2 = Closed Loop
3 = Servo Closed Loop
4 = Closed Loop-Position maintenance
Example: EM2 (closed loop)</td><td><n>EMi</td></tr>
</table>

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartStep according to the choices made in the SETUP dialog boxes.

Command	Command Description and Application Examples	Syntax
ER	**Encoder Resolution** Where i is an even integer Example: ER2000 (2000 counts/rev)	**<n>ERi**
ET	**End of Travel Switch Polarity** Selects the polarity of the EOT (End of Travel) switches. i = 0 NORM OPEN i = 1 NORM CLOSED	**<n>ETi**
FE	**Following Error Limit** Example: FE1000 (axis one 1000 counts)	**<n>FEi**
FL	**Fault Line Polarity** Fixed in SmartStep.	**<n>FL0**
GR	**Units Ratio** Example: GR4:1 (4 motor revolutions per distance (DU) unit)	**<n>GRi:i**
HE	**Home Edge** 0 = Negative Edge, 1 = Positive Edge. Example: HE0 (positive)	**<n>HEi**
HF	**Home Final Direction** 0 = Negative direction, 1 = Positive direction. Example: HF1 (one positive)	**<n>HFi**
HM	**Homing Mode** i=0 Switch Only i=1 Switch Then Z Channel i=2 Z Channel Only	**<n>HMi**
HO	**Home Offset** Example: HO1.0 (axis one 1.0 distance unit)	**<n>HO±r**
HS	**Home Switch** 0 = Normally Closed, 1 = Normally Open Example: HS1 (uses a normally open home switch-this is the default setting)	**<n>HSi**
ID	**Input Definition** Example: ID:UUUUUUUU The first 8 inputs are unassigned. All 8 input states must be specified. See also: OD Note: The G (Registration) Command is only valid for Input 1.	**<n>IDaaaaaaaa**
IL	**Idle** i = 1 Enables Idle Mode i = 0 Disables Idle Mode	**<n>ILi**
IR	**Position Maintenance Deadband** Sets position maintenance deadband in motor steps Valid as a program command using an immediate parameter only (No variables).	**<n>IRi**
JA	**Jog Acceleration** Example: JA.01 (.01 - Units selected by AU command.)	**<n>JAr**
JE	**Jog Enable** 0 = Jog Disabled, 1 = Jog Enabled Example: JE1 (enabled)	**<n>JEi**

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartStep according to the choices made in the SETUP dialog boxes.

Command	Command Description and Application Examples	Syntax
JH	**Jog High Velocity** Example: JH5.0 (5 in units selected by VU command.)	**<n>JHr**
JL	**Jog Low Velocity** Example: JL1.5 (1.5 in units selected by VU command.)	**<n>JLr**
MD	**Motor Direction Reference** 0 = Positive direction, 1 = Negative direction Example: MD0 (one positive	**<n>MDi**
MH	**Motor Inductance** a = H (High Inductance) a = L (Low Inductance)	**<n>MHa**
MI	**Motor Current**	**<n>MIn**
MR	**Motor Resolution** Default: 36,000 steps/rev (fixed resolution in SmartStep)	**<n>MRi**
MT	**Motor Type** MT10 (fixed motor type in SmartStep)	**<n>MT10**
MV	**Maximum Velocity** Example: MV50.0 (axis one 50 in units selected by VU command.)	**<n>MVr**
OD	**Output Definition** Example: OD:PPPPPPPP All 8 outputs defined as Programmable outputs. All 8 output states must be specified. See also: ID	**<n>ODaaaaaaaa**
OE	**Output States on Event** Configures output states on an event specified by a. OPTO positions 9-16 are only definable if configured as an output using OP command. a = P (Power-Up) F (Fault) S (Stop / Kill) i = 0 Off 1 On X No Change Example: OEF,0001XX01XXXXXXXX	**<n>OEa,iiiiiiii**
OP	**OPTO Configuration** i = Input, O = Output Example: OP:IIIIOOOO First four configured as inputs, last four as outputs	**<n>OPIIII OOOO**
PG	**Position Maintenance Gain** Sets position maintenance correction gain. i=1 to 32,767	**<n>PGi**
PU	**Power-Up Program** Example: PU105 (Runs program number 105 on power-up.)	**<n>PUi**
PV	**Position Maintenance Max. Velocity** Sets position maintenance maximum correction velocity (Units specified by VU command).	**<n>PVi**

Serial Setup Commands

These are the commands that the *Application Developer* program uses to configure the SmartStep according to the choices made in the SETUP dialog boxes.

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>PW</td><td>Password
Up to 4 characters: a-z, A-Z, 0-9
Entering a dash (-) will clear the password. Skipping a parameter will leave the password unchanged (see examples below).

Examples using PW:
Example 1: PW4FT,Q12h
Set the OPRATR password to 4FT and the ADMIN password to Q12h
Example 2: PW,New
Changes ADMIN password to New; leaves OPRATR password unchanged
Example 3: PW-,-
Clears both the OPRATR and ADMIN passwords</td><td><n>PWaaaa, aaaa</td></tr>
<tr><td>RE</td><td>Rest
n = 1 (Enables Rest Mode)
n = 0 (Disables Rest Mode)</td><td><n>REi</td></tr>
<tr><td>SN</td><td>Scan Conditions
Conditions stopping program select line scanning are represented by “i”
<table><tr><td></td><td>ESC</td><td>Stop</td><td>Limit +</td><td>Limit -</td><td>Kill</td><td>Fault</td><td>Interrupt</td></tr><tr><td>SN</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td></tr></table>0 = Continue program select scanning,
1 = Stop program select scanning on this condition
Example: SN:0111111 (all input conditions, except pressing the ESC key stop program select line scanning)</td><td><n>SNiiiiiiii</td></tr>
<tr><td>SR</td><td>Stop Deceleration Rate
Example: SR100 (100 rps^2)
Note: Stop Deceleration Rate units are always in rps^2 and are not user selected</td><td><n>SRr,r</td></tr>
<tr><td>UN</td><td>Unit Number
Example: UN5 (sets unit address to 5)</td><td><n>UNi</td></tr>
<tr><td>VU</td><td>Velocity Units
<table><tr><td>0</td><td>units/sec (where “units” is a string defined by the DU command)</td></tr><tr><td>1</td><td>units/min (where “units” is a string defined by the DU command)</td></tr><tr><td>2</td><td>rps</td></tr><tr><td>3</td><td>rpm</td></tr></table>Example: VU2 (axis one set to rps)</td><td><n>VUi</td></tr>
<tr><td>WA</td><td>Waveform
i = 1 (-4% 3rd harmonic)
i = 0 (SINUSOID)</td><td><n>WAi</td></tr>
</table>

Serial Programming Commands

The commands in this category may be sent to the SmartStep'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 SmartStep's volatile memory. See *Chapter 6, Programming Commands* for complete definitions.

Command	Command Description and Application Examples	Syntax
AC	**Acceleration** Example: AC10	**ACr**
DA	**Distance Absolute** Example: DA15	**DAr**
DC	**Distance to a Change** Example: DC20	**DC±r**
DE	**Deceleration** Example: DE.2	**DE±r**
DI	**Distance Incremental** Example: DI 3.42	**DI±r**
EA	**Enable Axis** i=0 (disable drive), i=1 (enable drive), i = 2 (standby) Example: EA0 disables axis	**EAi**
GH	**Go Home** Example: GH10	**GH±r**
GI	**Go Immediate**	**GI, GIi**
GO	**Begin Move**	**GO, GOi**
MC	**Move Continuous**	**MC+**
OT	**Set Outputs** Example: OT5,101 (turns on outputs 5 and 7, turns off output 6)	**OTi,iii...** **OTiii...**
" "	**Send String Over Serial Port** Example: GO "End of Move" (sends "End of Move" out serial port after move)	**"user text"**
RG	**Registration** Example: RG3.5	**RGr**
SP	**Set Position** Example: SP15.0 (set axis position to 15.0)	**SPr**
SQ	**Square Root** Example: SQ16, (x)	**SQr,(var)**
ST	**Stop on Input** Example: ST4 DI50 GO (motor will decelerate to a stop or remain stopped if input 4 is activated). ST0 disables the input	**STi**
TD	**Time Delay** Example: OT1,1 TD.12 OT1,0 Turn Output 1 on for 120 msec.	**TDr**
VE	**Velocity** Example: VE50 (sets speed for the next move to 50, in units set by the VU command)	**VEr**

Serial Programming Commands

The commands in this category may be sent to the SmartStep'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 SmartStep's volatile memory. See *Chapter 6, Programming Commands* for complete definitions.

Command	Command Description and Application Examples	Syntax
WT	**Wait On Input Condition** Example: WT1,0 GO (wait for input 1 to turn off before starting move)	**WTi,ii...** **nWTiii...** **<n>WT expression** **<n>WT#i**

Commands Not Available In Hosted Mode The following set of commands can only be executed if they are part of, or within, a program:	
Command	**Command Name**
BR	**Break**
EB	**End Block**
GS	**Go Sub**
GT	**Go To**
IF	**If**
IV	**Input Variable**
LP	**Loop**
MS	**Display Message**
EN	**End Routine**
FK	**Function Key**
LU	**Loop Until**
LW	**Loop While**
WT	**Wait**

Serial Immediate Status Commands

Serial Immediate Status commands are processed immediately upon receipt, rather than waiting in the buffer for previous commands to finish. They can be issued while a program is running, or while motion is in progress. They cannot be stored within a program.

Using Immediate Status Commands

Serial Immediate Status commands are provided for two purposes. One is to allow a host control to query the SmartStep in real time, for system, position, and I/O status. The second is to provide a means to perform in-depth troubleshooting via RS-232C. These commands will interrupt the SmartStep and generate a return. They do not affect operation of the SmartStep.

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 SmartStep by this host device, such as a computer or PLC. The following commands are provided so the host can verify the status of the SmartStep before commanding motion. The System Status (SS) command returns overall system information, and indicates general faults. The Axis and Drive (SAi and SDi) commands can then be used to provide more detailed, axis-specific information.

These commands are also an invaluable system troubleshooting aid. Since they are immediate commands, they will generate a response from the SmartStep even if it is in the middle of move, waiting for an input condition to become true, etc. Checking the System Status and the I/O Status will provide enough information to explain what the SmartStep is doing. If a fault is indicated, the Drive Status and Axis Status commands can give detailed, axis-specific information.

Summary of Immediate Status Commands

Command	Syntax	Purpose
Clear Command Buffer	<n>CB	Clears the terminal input buffer and buffered command buffer
Input States	<n>IS	Real-time status of discrete and OPTO inputs
Kill	<n>K	Issues immediate halt to current and programmed motion
Model Number	<n>MN	Returns unit model number over RS-232C
Output States	<n>OS	Real-time status of discrete and OPTO outputs
Current Position	<n>PA1	Real-time position, in user units, of axis 1
Stop	<n>S	Issues program terminaiton, decelerates to a halt
Axis Status	<n>SA1	Returns axis specific status (i.e. limit and home states) of axis 1
Drive Status	<n>SD1	Returns drive specific status (i.e. type of amp fault) of axis 1,
System Status	<n>SS	Returns general system status and operation

Serial Immediate Status Commands

Note: All but the S and K commands require an address

<table>
<tr><th>Command</th><th>Command Description and Application Examples</th><th>Syntax</th></tr>
<tr><td>CB</td><td>Clear Command Buffer
Clears the terminal input buffer and buffered command buffer</td><td><n>CB</td></tr>
<tr><td>IS</td><td>Tell Input States
Returns the current state (on or off) of the 8 inputs. The status is returned as a four digit hexadecimal number, preceded by an asterisk. The least significant digit represents the binary value of inputs 4-1.

Example: IS returns *00F6<cr> with the input conditions shown in this table.
<table>
<tr><td colspan="8">n/a</td><td colspan="8">Inputs</td></tr>
<tr><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>on</td><td>on</td><td>on</td><td>on</td><td>off</td><td>on</td><td>on</td><td>off</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">F</td><td colspan="4">6</td></tr>
</table>
Your computer program must decode the hexadecimal number to determine the state of each input.</td><td><n>IS</td></tr>
<tr><td>K</td><td>Kill
Issuing the K command causes the control to abruptly stop commanding further motion and 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 sytems mechanics. The Stop commands provides a more controlled halt.</td><td><n>K</td></tr>
<tr><td>MN</td><td>Model Number
Returns the unit model number.
MN command responses are: *SmartStep *SmartStep23 *SmartStep240</td><td><n>MN</td></tr>
<tr><td>OS</td><td>Tell Output States
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 *00F6<cr> with the Output conditions shown in this table.
<table>
<tr><td colspan="8">n/a</td><td colspan="8">Inputs</td></tr>
<tr><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>off</td><td>on</td><td>on</td><td>on</td><td>on</td><td>off</td><td>on</td><td>on</td><td>off</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">F</td><td colspan="4">6</td></tr>
</table>
Your computer program will have to decode the hexadecimal number to determine the state of any output.</td><td><n>OS</td></tr>
<tr><td>PA</td><td>Tell Absolute Position
Reports current position in user units based on encoder mode selected. Can report specifically commanded or encoder position when PAa,n is used.

Example: PA1 returns *+1.000 (the position of axis one)</td><td><n>PA1</td></tr>
</table>

Serial Immediate Status Commands Note: All but the S and K commands require an address		
Command	**Command Description and Application Examples**	**Syntax**
RS	**Reset System** Re-initializes, or warm boots, the control software to its power-up state. The initialization process takes about 10 seconds to complete. Programs and configuration settings are not erased. This command is the equivalent of cycling power.	**<n>RS**
S	**Stop** 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.	**<n>S**

Serial Immediate Status Commands

Note: All but the S and K commands require an address

Command	Command Description and Application Examples	Syntax
SA	**Tell Axis Status** Returns the current axis status as a four digit hexadecimal number, preceded by an asterisk. Your controller program will decode the hexadecimal number to determine the axis status. Example: SA1 returns *002A<cr>. This means Axis 1 is not moving, the last move completed successfully, and the home switch is on.	**<n>SA1**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	on	off	on	off	on	off
0				0				2				A			

Description	bit #	Parameter Definition
Not Moving/Moving	1	1=Steps being sent to the amplifier 0= No steps being sent
At Velocity	2	1= Stepping at a constant rate (includes zero velocity) 0= Step rate is changing
In Range	3	B8961/2 only
Move Command Complete (Same as Move Done Output)	4	1=The correct number of steps were sent without an amp fault, following error, or hitting an End of Travel limit. 0=Reset to zero at the beginning of each move.
Home Successful	5	1= The last homing move was successful 0= At power up, reset to zero at the start of the next jog, GO, or GH.
Home Switch Status	6	Hardware status of home switch. 0=off, 1= on
- Limit Switch Status	7	Hardware status of limit switch 0=off, 1= on, limits require a NC switch
+ Limit Switch Status	8	Hardware status of limit switch 0=off, 1= on, limits require a NC switch
- Limit Switch Latched	9	1= Set when a move is terminated by a limit in the - direction. Cleared at the start of a move in the + direction. 0= At power up or reset, even if on the - limit.
+ Limit Switch Latched	10	1= Set when a move is terminated by a limit in the + direction. Cleared at the start of a move in the - direction. 0= At power up or reset, even if on the + limit
RESERVED	11	State undefined, should be masked
RESERVED	12	State undefined, should be masked
RESERVED	13	State undefined, should be masked
RESERVED	14	State undefined, should be masked
RESERVED	15	State undefined, should be masked
RESERVED	16	State undefined, should be masked

Serial Immediate Status Commands

Note: All but the S and K commands require an address

Command	Command Description and Application Examples	Syntax
SD	**Tell Drive Status** Returns the current drive status as a four digit hexadecimal number, preceded by an asterisk. Your controller program decodes the hexadecimal number to determine the drive status. Example: SD1 returns *0010<cr>. This means Axis 1 is enabled, in position mode, and is not faulted.	**<n>SD1**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	off	on	off	off	off	off
0				0				1				0			

Description	bit #	Parameter Definition
Following Error	1	1= Following error occurred 0=At power up and reset. Set to zero at the start of the next move.
Over-current	2	1= Over Current (requires reset to clear) 0= At power up and after reset.
Thermal Fault	3	1= Thermal fault in the motor or drive (requires reset to clear) 0= At power up and after reset.
RMS Over-current	4	*B8961/2 only*
Drive Enabled	5	1= Enable Drive (see also EA1) 0= Disable Drive (see also EA0)
RESERVED	6	State undefined, should be masked
RESERVED	7	State undefined, should be masked
Torque/Position	8	*B8961/2 only*
Amplifier Fault	9	1= The amplifier is faulted. Requires a power cycle to reset. 0= At power up or reset
RESERVED	10	State undefined, should be masked
RESERVED	11	State undefined, should be masked
RESERVED	12	State undefined, should be masked
RESERVED	13	State undefined, should be masked
RESERVED	14	State undefined, should be masked
RESERVED	15	State undefined, should be masked
RESERVED	16	State undefined, should be masked

Serial Immediate Status Commands

Note: All but the S and K commands require an address

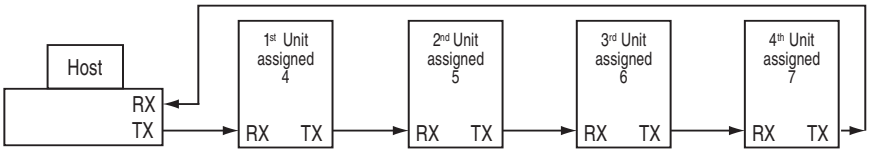
Command	Command Description and Application Examples	Syntax
SS	**Tell System Status** Returns the current system status as a four-digit hexadecimal number, preceded by an asterisk. Your controller program decodes the hexadecimal number to determine the system status. Example: SS returns *0001<cr> means there are no amplifier faults, and no programs running - SmartStep is ready to process any buffered RS-232C command.	**<n>SS**

Status															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
off	off	off	off	off	off	off	off	off	off	off	on	off	off	off	on
0				0				0				1			

Description	bit #	Parameter Definition
Ready to Buffer RS-232C Commands	1	1= Ready to buffer RS-232C commands 0= Initializing from a power-up or reset, or unchecked errors exist. Any buffered commands sent will be discarded.
FLASH Error	2	1= Non-volatile memory checksum error, all programs were deleted on power up. 0= Non-volatile memory checksum OK
Program Running	3	1= Running a program 0= Not running a pre-defined program.
FK Active	4	1= Paused waiting for a function key. 0= Not waiting at a FK command.
WT Active	5	1= Paused waiting for a input condition. 0= Not waiting at a WT command
TD Active	6	1= Paused at a time delay. 0= Not waiting at a TD command
Waiting for IV	7	1= Paused, waiting a variable input. 0= Not waiting at a IV command
Buffer Full	8	1= RS-232C buffer 75% full Total Capacity: 2k 0= RS-232C buffer less than 60% full.
Axis #1 Fault	9	1= Amp fault, following error, move stopped by limit switch (see SAi and SDi for more detailed fault information) 0= No faults
Axis #2 Fault	10	1= Amp fault, following error, move stopped by limit switch (see SAi and SDi for more detailed fault information) 0= No faults
RESERVED	11	State undefined, should be masked
Program Select Scanning	12	1=BCD and Binary program select scanning enabled. 0=A Stop Scan condition has occurred or no inputs are configured as program select lines.
Data Download Status	13	1=Data Transfer failed (program memory overflow) 0=Data successfully received
RESERVED	14-16	State undefined, should be masked

Serial Supervisory Commands

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

Command	Command Description and Application Examples	Syntax
AA	**Auto Address** The AA command automatically addresses SmartStep 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*i*.	**<n>AA or AAi**
DP	**Delete Program** Erases a program from memory, where i is the program number. This is equivalent to pressing the delete key on the keypad and entering the program number. Range: 1-199 (1-400 with 30K memory option) Example: DP99 (deletes program number 99).	**<n>DPi**
DR	**Download Program to RAM** Begins downloading a program from the host to the control's RAM, rather than non-volatile memory. (Also see the PR command description) These programs will be lost after a reset or power cycle. The program string must end in EP. The commands between DR and EP do not need a device address. The DR command is typically used when the control is operated exclusively via a host controller which constantly downloads and executes programs. This increases the usable life of the FLASH. Range: 1-199 (1-400 with 30K memory option) Example: 1DR50 AC4 DE4 VE30 LP6 DI10.5 GO EN 1EP RN50 Downloads program #50 to Unit #1's RAM, then runs program #50	**<n>DRi**
EC	**RS-232C Echo Enable/Disable** 0 = echo Disabled, 1 = echo Enabled. Example: EC0 (echo off) The RS-232C Echo must be enabled for daisy chain operation	**<n>ECi**
EP	**End Program Definition** Denotes the end of a program definition. All program definitions must begin with nPRi or nDRi and end with EP. Example: PR15 [part A] AC4 VE30 DI10.5 GO EP	**<n>EP**

Serial Supervisory Commands

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

Command	Command Description and Application Examples	Syntax
EX	**Ends Upload All or Load All** Singles the end of a upload all (UA) or load all (LA) sequence. EX is sent by the SmartStep to the host after completing a UA. EX is sent by the host to the SmartStep to terminate a LA.	**<n>EX**
LA	**Load All** Sent to the SmartStep before downloading a long list of setup parameters and programs. This command will disable the non-addressed units so that each setup parameter doesn't need an address. Must be followed by an EX to reestablish the daisy chain communications.	**<n>LA**
LS	**List Programs** Lists number of programs, memory usage, and the current available memory of the SmartStep. Just like **Edit/List** from the keypad.	**<n>LS**
OC	**Original Configuration** Returns the FLASH to its original factory-default state. The command buffer is cleared, all programs are erased, and all configuration settings are returned to their default values	**<n>OC**
PR	**Define Program** Starts a program definition. Just like the DR command, but writes the Smart-Steps 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).	**<n>PRi**
RN	**Run Program** This commands any program, by number only. The RN command does not support the optional program names. Example: RN25	**<n>RNi**
SW	**Tell Software Version** The control returns its software revision. **Example: 1SW returns *V1.40 <cr>**	**<n>SW**
UA	**Upload All** Uploads all setup parameters and programs from unit n. SmartStep sends an EX to terminate upload.	**<n>UA**
UL	**Upload Program** Range: i=1-199 (program #) (up to 400 programs with 30K option) Uploads program number i to the host - SmartStep adds brackets. Example: 1UL2 Uploads program 2 from unit #1. Response: { [part A] AC4 VE30 DI10.5 GO }	**<n>ULi**

Chapter 9 - Hardware Reference

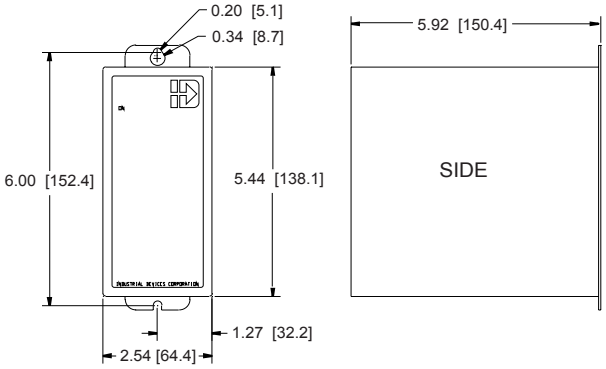
Mounting Your SmartStep

The standard mounting bracket on the SmartStep has been integrated with the housing to better facilitate minimum-width mounting. A minimum-depth mounting bracket is available as SmartStep-MD.

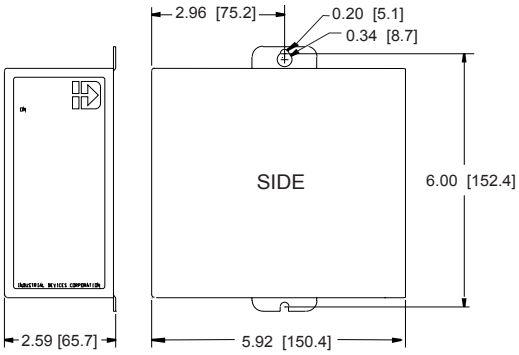
These general mounting guidelines should be observed:

- The vertical clearance between a SmartStep and other equipment or surfaces of the enclosure (including other SmartSteps) should be a few inches on top and bottom to allow for power and motor connections.
- Horizontal clearance is not required between SmartSteps, but we recommend 0.05'' to better facilitate handling of the units.

Typical Mounting Fasteners	
Cap Screw	Machine Screw
#10, #8, M4	#8, M4



Mounting for Minimum Width Configuration (Standard)



Mounting for Minimum Depth Configuration (SmartStep-MD Option)

This page intentionally left blank.

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.

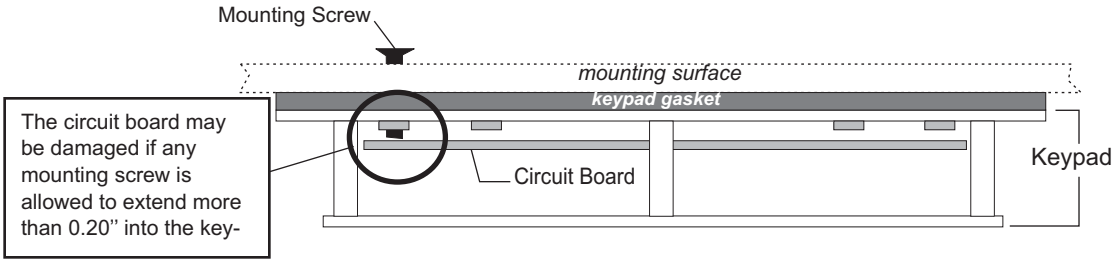
An FP220 Keypad Mounting Template is included with every keypad, and a *not to scale* mounting dimensions drawing can be found on the next page for reference. Please pay particular attention to the **CAUTION** on the template.

Warning

1) When mounting the keypad on wall-type surfaces (as shown in the illustration below) be absolutely certain that you have accurately estimated the proper length of mounting screws. Your keypad may be damaged if a mounting screw is allowed to extend more than *0.20 inches* into the keypad.

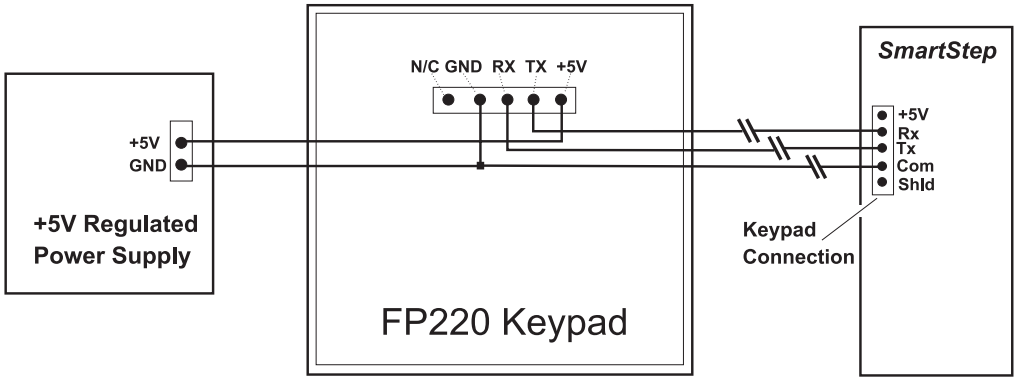
2) When mounting the keypad to a metal surface, it is highly advisable that the keypad be insulated from the metal surface by using non-conductive fasteners (screws, washers, etc.).

End View of Wall-Mounted Keypad with Cover Removed



Extending the Cable Length to Your Keypad

The keypad cable may be extended if necessary, but at longer distances your keypad may require a separate, regulated 5 VDC (500mA) power supply as shown in the illustration below. Consult IDC Applications Engineering if you have questions about this procedure.

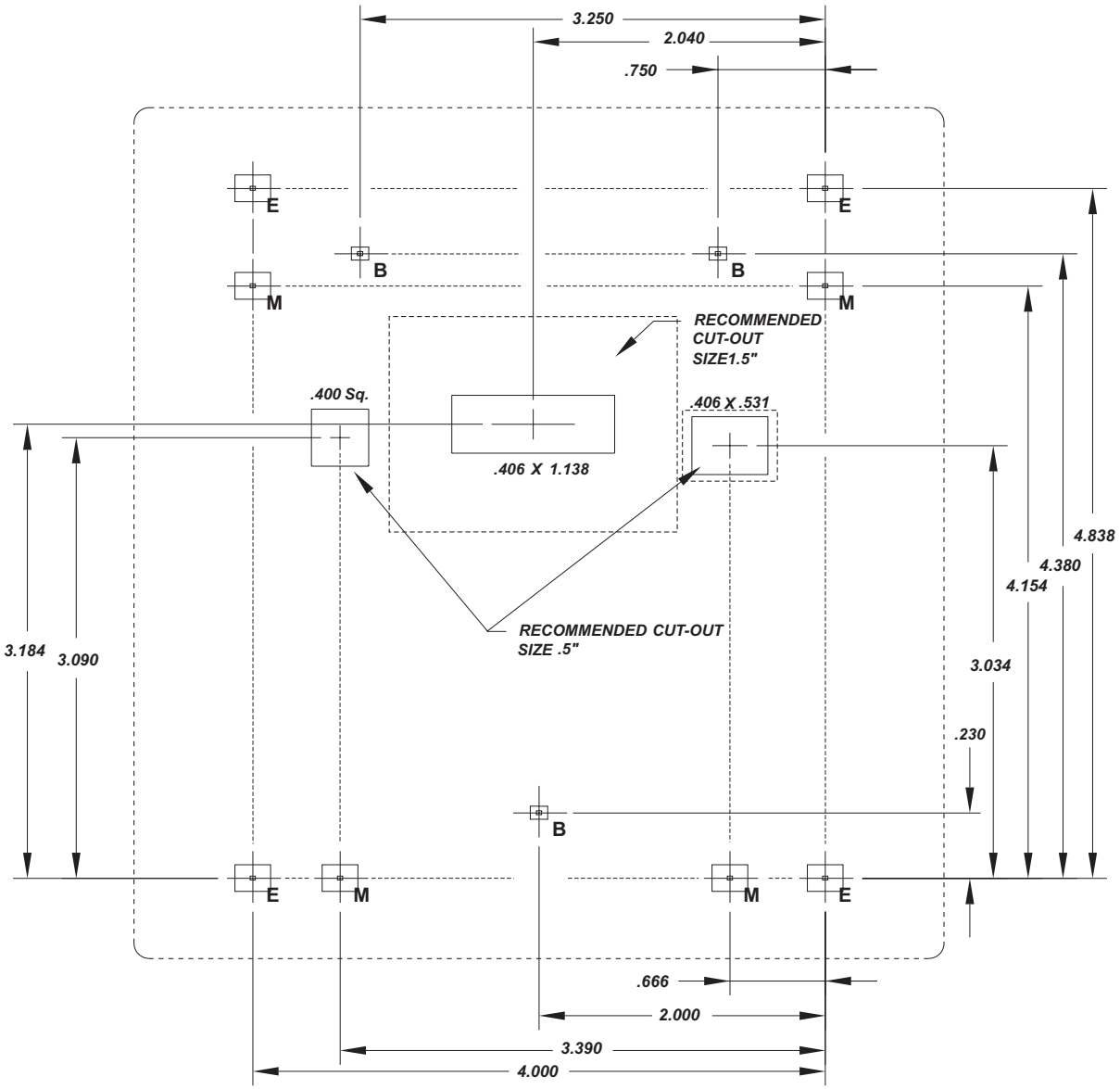


FP220 Keypad Mounting *Template

***CAUTION** - this is *scaled-down version* of the keypad template. Use this drawing only for dimensions and locations of mounting holes. The actual template is included with your keypad and may also be found on the IDC documentation CD.

CAUTION: ***Your Keypad will be damaged if mounting screws extend more than 0.2 inches into the keypad.***

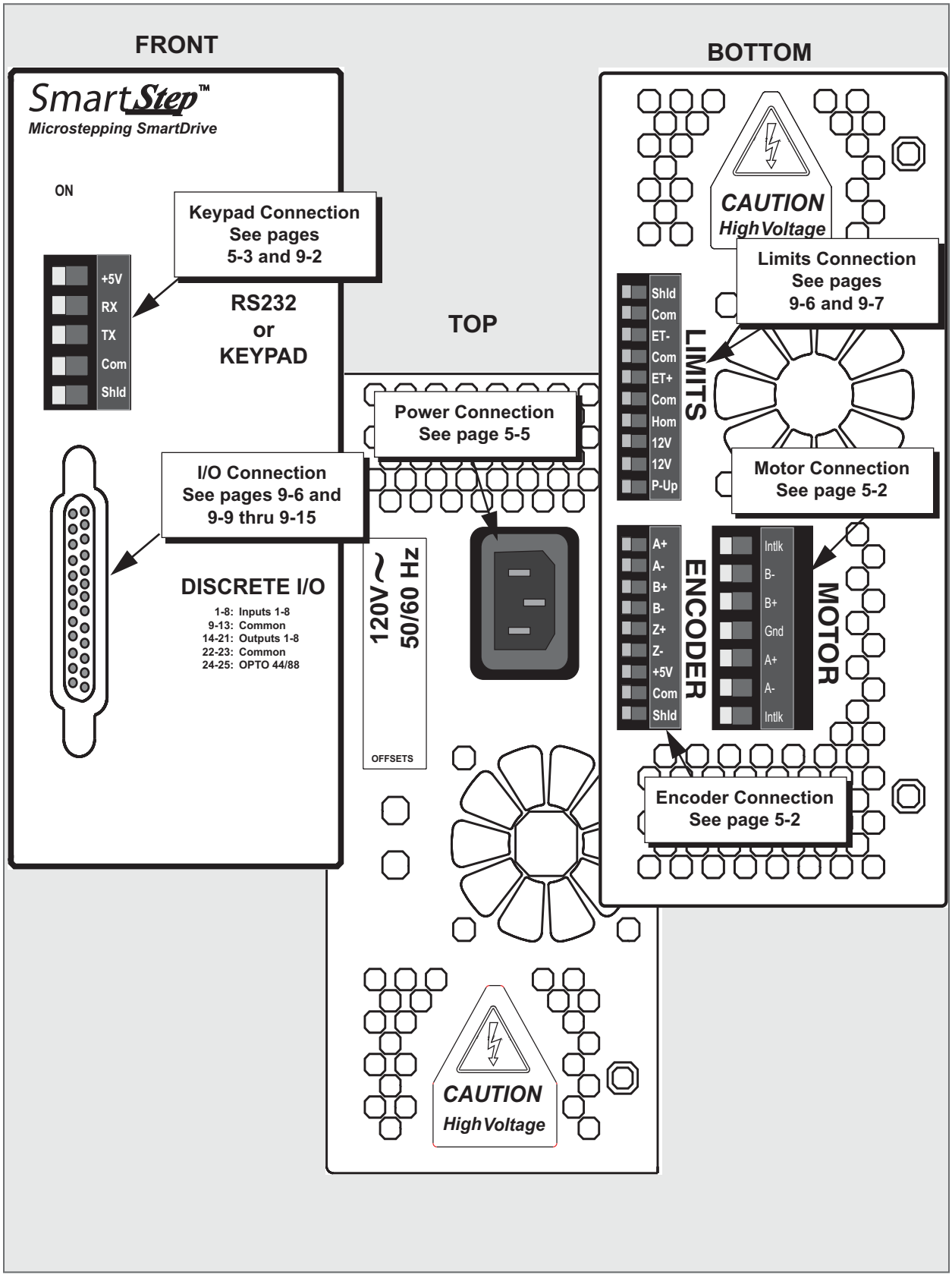
E	3/16"CLEARANCEHOLES(4)	CUT THESE HOLES FOR 6/32 CAPTIVE MOUNTING
M	3/16"CLEARANCEHOLES(4)	CUT THESE HOLES FOR M 3.5 CAPTIVE MOUNTING
B	5/32"CLEARANCEHOLES(3)	CUT THESE HOLES FOR BALL-HEAD REMOVABLE MOUNTING



SmartStep Specifications

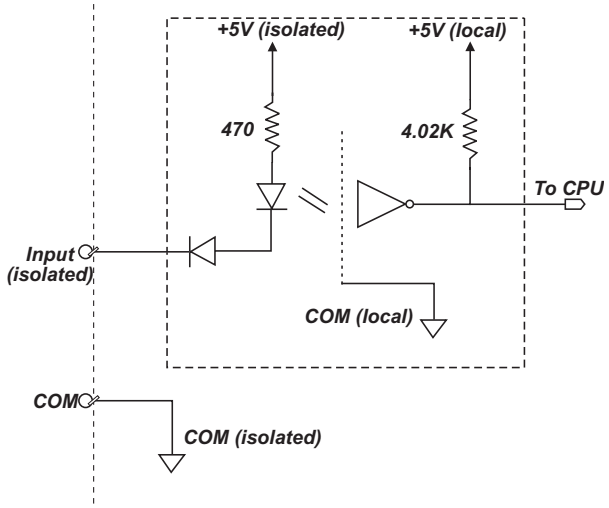
Input Power Requirements	SmartStep	90-120 VAC single phase, 50/60 Hz, 500 VA max @ 7.9 amp setting
	SmartStep 240	190-240 VAC single phase, 50/60 Hz, 500 VA max @ 3.9 amp setting
	SmartStep 23	90 - 120 VAC single phase, 50/60 Hz, 250 VA max @ 3.0 amp setting
Output Power Available	12 VDC internal power supply, 250 mA maximum output current	
	5 VDC at 200 mA available for encoder.	
Environmental Requirements	Ambient Temperature	0 - 50˚C
	Humidity	0% - 90% non-condensing
Drive Signals	Position Range	± to 2,147,483,647 steps (absolute and incremental)
	Velocity Range	.0002 - 70 rps
	Acceleration Range	1 to 20,000,000 steps/sec^2
Encoder Interface	Optically isolated, differential 5 VDC, 2 MHz maximum (post-quadrature)	
Outputs	8 programmable outputs. Open collector, sink current - 100 mA maximum	
Inputs	8 programmable inputs + 2 Limits + Home 24 VDC maximum - optically isolated 3 mA current sinking is required at voltage no greater than 0.7 volts	
Programming	IDeal programming language. Program from the keypad operator panel, or via your PC using IDC's Application Developer™ software (included).	

SmartStep Hardware Connections

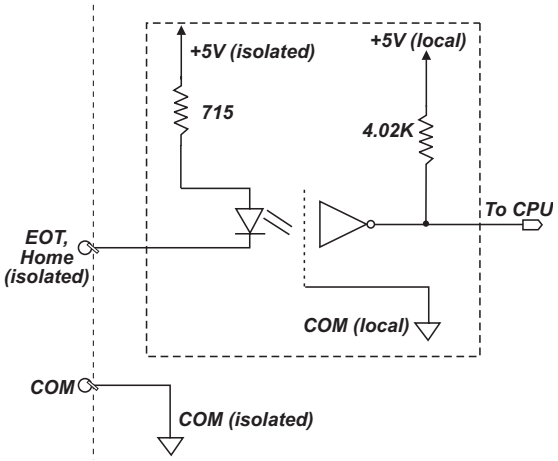


SmartStep Schematics

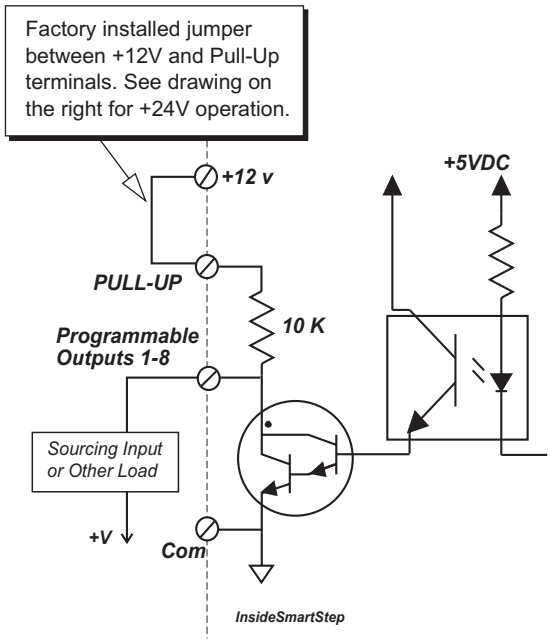
Discrete Inputs



Home & End-of-Travel Inputs

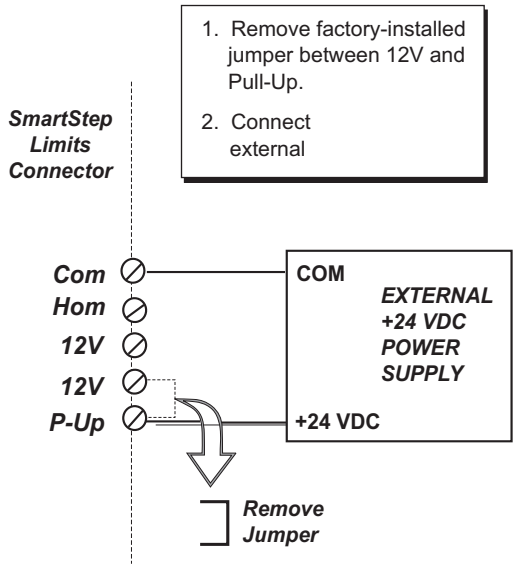


Programmable Outputs

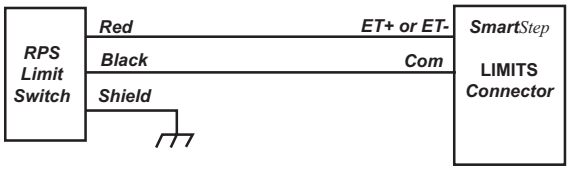
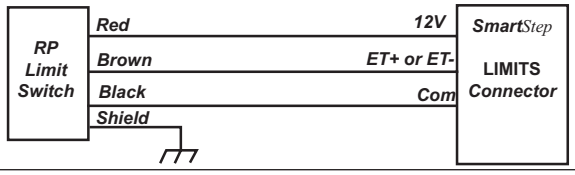
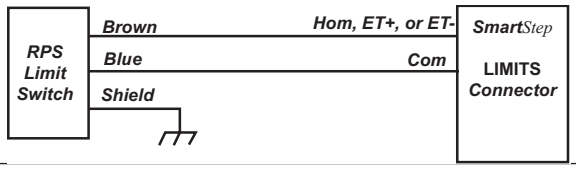
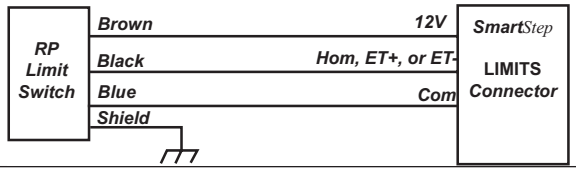


Max. sink current: 100 mA per output
Total 350 mA available from 12VDC supply

Connecting a +24V Power Supply



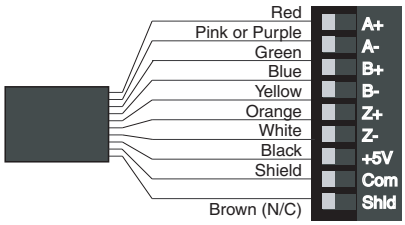
Connecting IDC Limit Switches to the SmartStep

IDC Limit Switches			
Switch	Type	Comments	Connections
RPS-1 RPS-2	Normally Open Home only Normally Closed	Reed Switch - less wiring and lower cost Mounts to N, T, R2, R3, and R4 Actuators	
RP-1 RP-2	Normally Open Home only Normally Closed	Hall Effect Switch - has longer life, and uses LEDs Mounts to N, T, R2, R3, and R4 Actuators	
PSR-1 PSR-1Q PSR-2 PSR-2Q	Normally Open Home Only Normally Closed	Reed Switch - "Q" indicates Quick Disconnect Mounts to EC, NV	
PSN-1 PSN-1Q PSN-2 PSN-2Q	Normally Open Home Only Normally Closed	NPN Hall Effect - "Q" indicates Quick Disconnect Mounts to EC, NV	
PSP-1 PSP-1Q PSP-2 PSP-2Q	Normally Open Normally Closed	PNP Hall Effect Used on Controls requiring PNP Mounts to EC, NV	*Will Not Work with IDC Controls*

Connecting an Encoder to a SmartStep

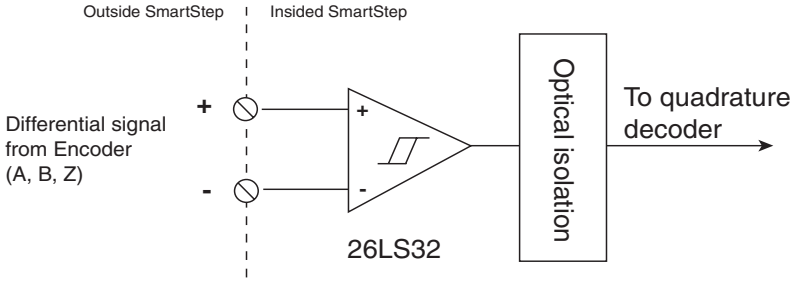
Color codes are for IDC encoders only. Use signal names for other manufacturer's encoders.

Encoder Wiring

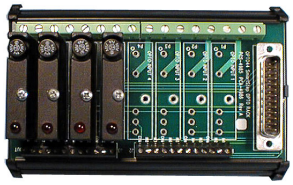
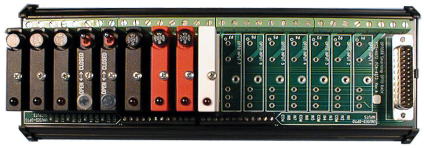
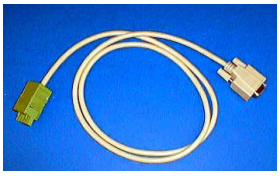
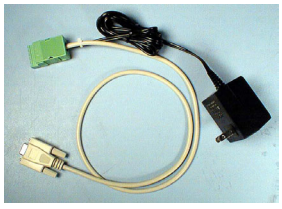


Connector on SmartStep

Encoder Input Schematic

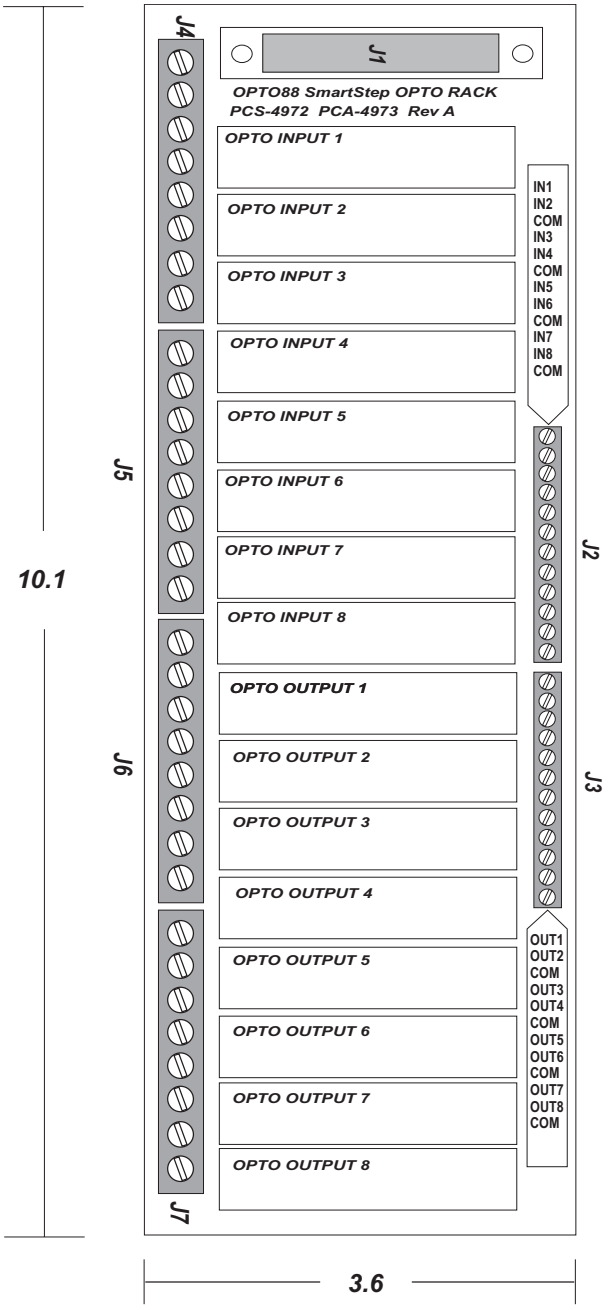


SmartStep Accessories

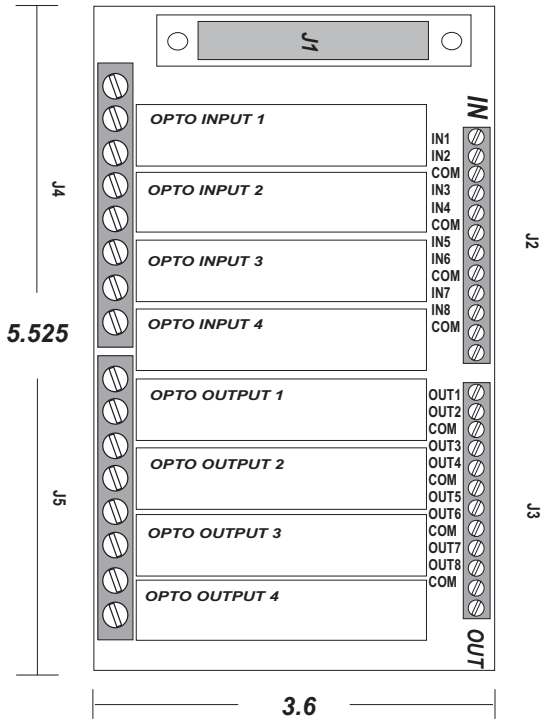
SmartStep I/O Accessories		
Accessory (P/N)		**Description**
OPTO44		OPTO Rack - accepts up to 8 optional conditioning modules. (See Opto Module table in this chapter for list of modules available from IDC). See the following page for dimensions.
OPTO88		OPTO Rack - accepts up to 16 optional conditioning modules. (See Opto Module table in this chapter for list of modules available from IDC). See the following page for dimensions.
DB25BO	 Dimensions: 2.0L x 1.0W x 0.8D	Screw Terminal Breakout Board See page 9-14 for connection information.
SS-PNP-BO	 Dimensions: 2.0L x 1.5W x 1.28D	Screw Terminal Breakout Board to Convert to Sourcing Outputs See page 9-15 for connection information.
SS-IO SS-IO-6		I/O cables that connect SmartStep to other devices or PLC SS-IO is 2 ft. SS-IO-6 is 6 ft. See page 9-13 for wire color codes.
SS-RS232		Cable for connecting SmartStep to PC (9-pin Comm. Port)
PCS-5004		PC-Keypad Cable for copying programs between keypad and PC

OPTO Racks

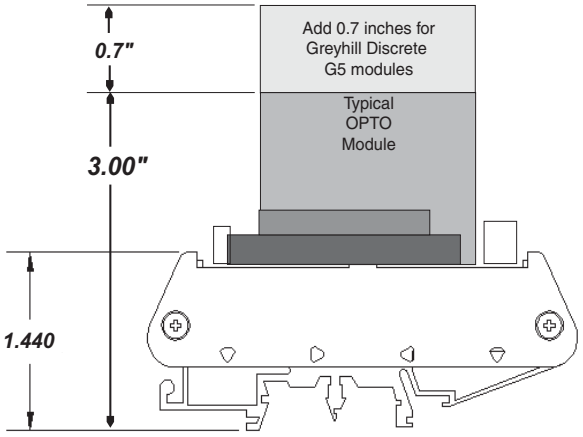
OPTO88 RACK



OPTO44 RACK



See page 9-10 and 9-11 for important information on making connections to the OPTO44 and OPTO88 Racks.



Making OPTO Rack Connections

OPTO44 and OPTO88 Racks allow the continued use of discrete inputs and outputs while providing the added dimension of opto-conditioning. With this added dimension comes an important **CAUTION** that must be observed when using either of these OPTO Racks.

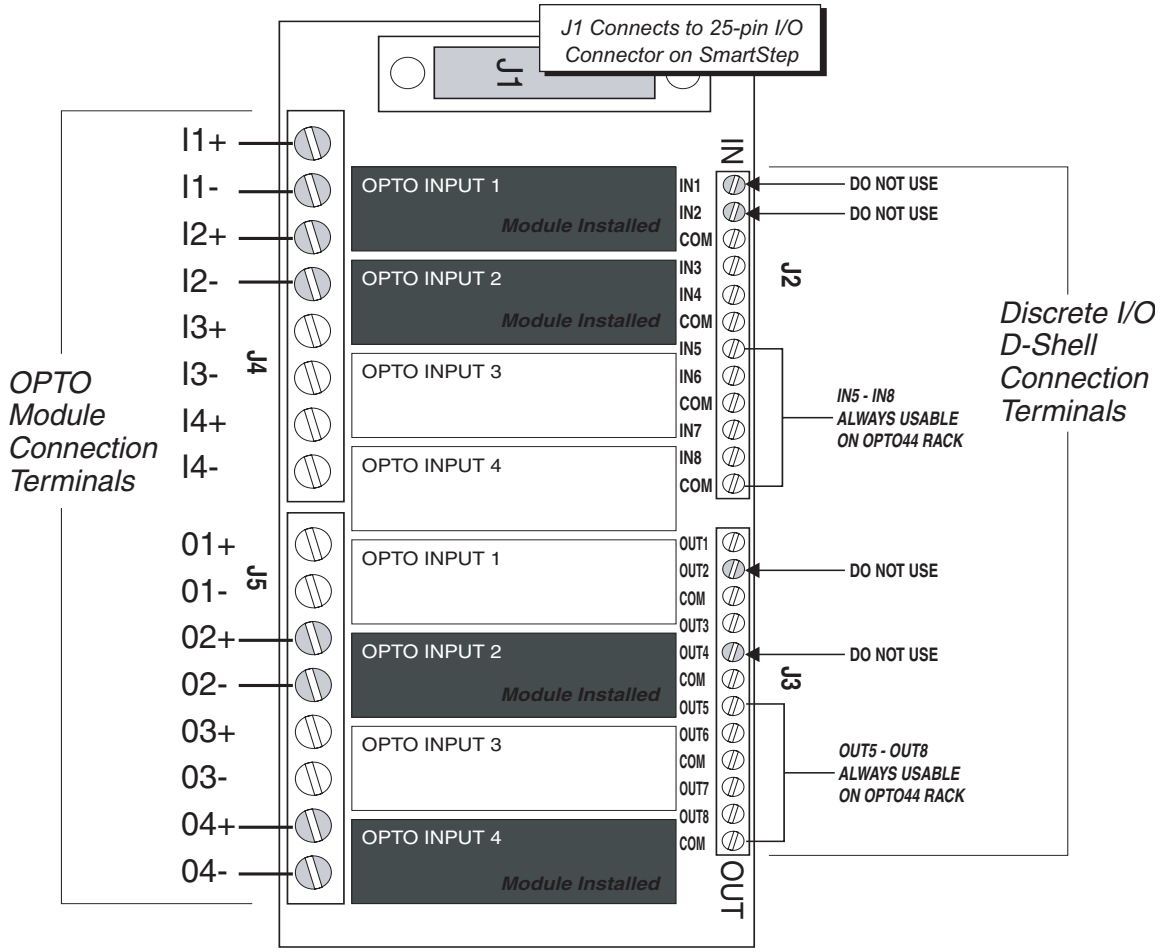
OPTO44 Connections

CAUTION - Do not use discrete inputs IN1 - IN4 and discrete outputs OUT1 - OUT4 if a corresponding OPTO module is being used (I1 - I4 and O1 - O4). Failure to observe this caution may result in damage to system components.

Discrete inputs IN5 - IN8 and discrete outputs OUT5 - OUT8 may be used anytime because there are no corresponding OPTO positions.

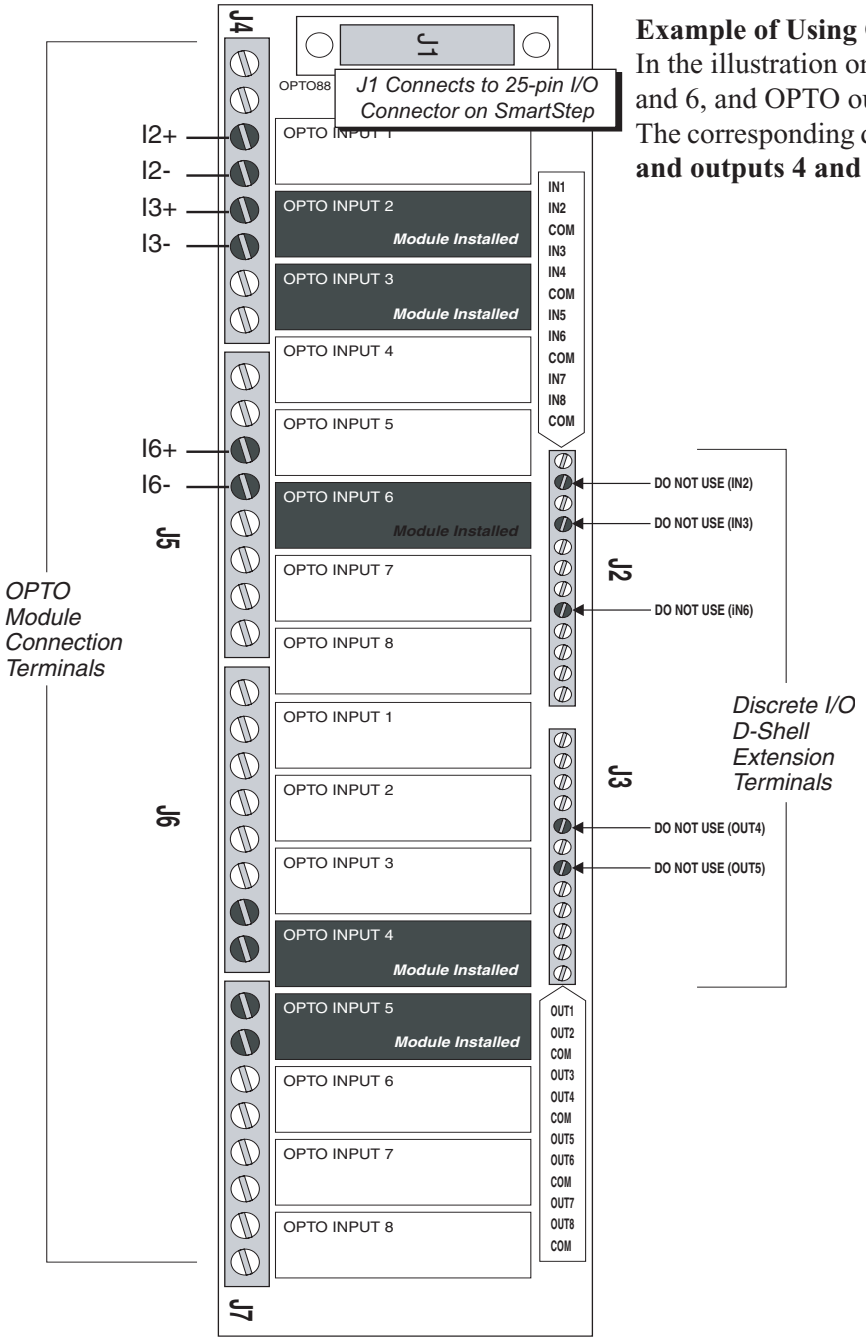
Example of Using OPTO44 Discrete I/O

In the illustration below, OPTO inputs 1 and 2 and OPTO outputs 2 and 4 are being used. In this case, discrete inputs IN1 and IN2 and discrete outputs OUT2 and OUT4 must not be used.



OPTO88 Connections

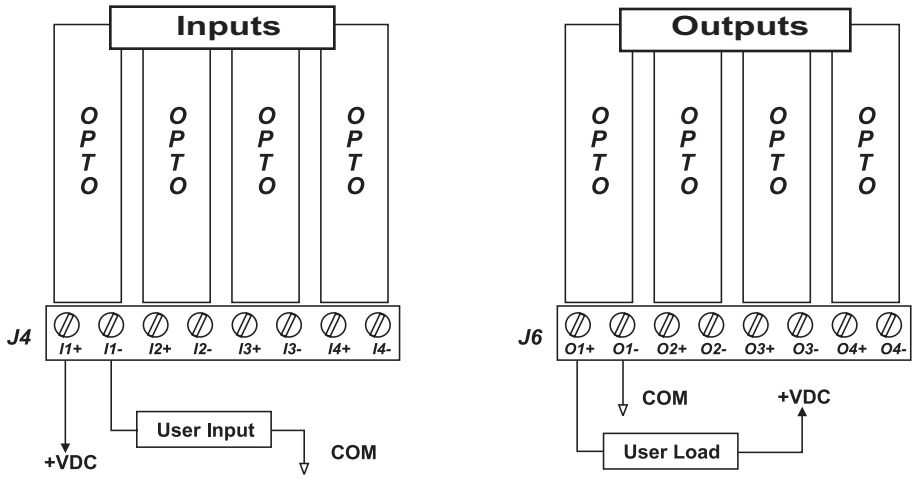
CAUTION - Do not use <u>any</u> discrete input or output for which a corresponding OPTO module is installed. The OPTO44 Rack allows usage anytime of discrete I/O IN5-IN8 and OUT5-OUT8, but this does not apply to the OPTO88 Rack. Failure to observe this caution may result in damage to system components. See example and illustration below.



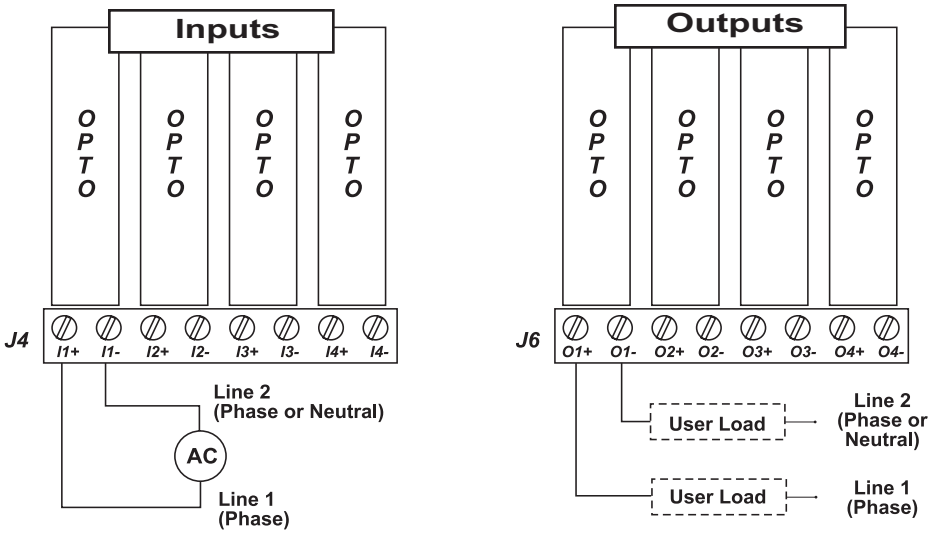
Example of Using OPTO88 Discrete I/O
In the illustration on the left, OPTO inputs 2, 3, and 6, and OPTO outputs 4 and 5 are being used. The corresponding discrete I/O, **inputs 2, 3, and 6, and outputs 4 and 5, must not be used.**

Using OPTO44 and OPTO88 - Wiring Examples

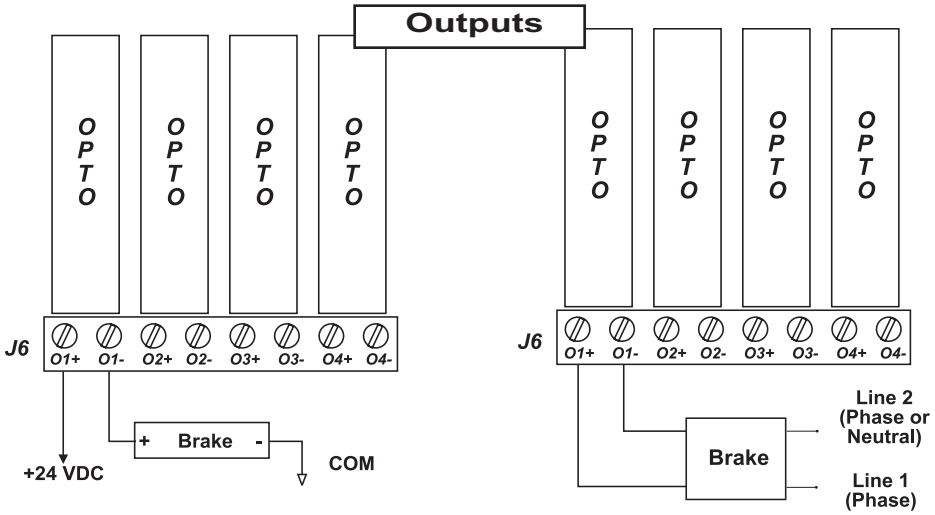
DC I/O



AC I/O



Brake Outputs



OPTO Modules Available from IDC

IDC stocks the following OPTO modules, which may be specified when ordering a SmartStep:

OPTO MODULES			
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)	Analog Input Module		73G-IV10
K (PCB-1220)	Analog Input Module		73G-II420

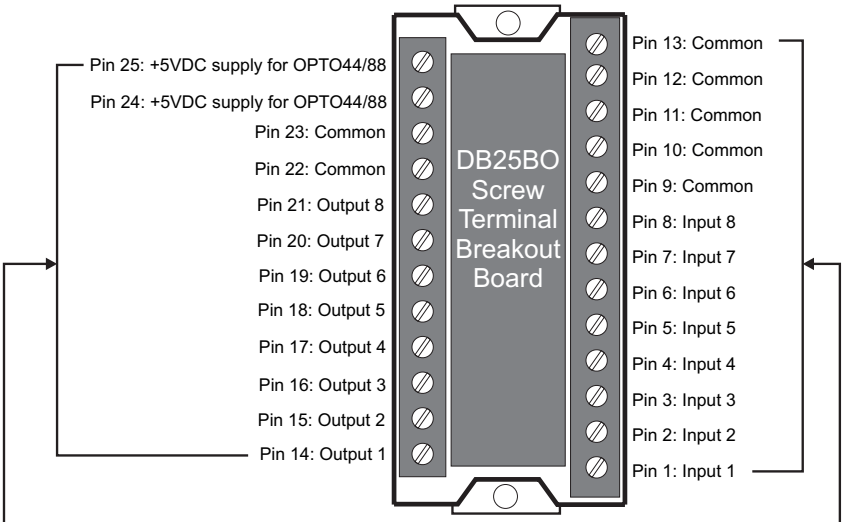
More information on these OPTO modules is available from the OPTO module manufacturer or your local distributor.

Wire Color Codes for Optional SS-IO and SS-IO-6 Cables

Color Codes for SS-IO and SS-IO-6 Cables									
Pin#	Wire Color	**Pin#**	Wire Color	**Pin#**	Wire Color	**Pin #**	Wire Color	**Pin#**	Wire Color
1	Brown	**6**	Green	**11**	White	**16**	White/Green	**21**	Black/Orange
2	Red	**7**	Light Green	**12**	Black	**17**	White/Blue	**22**	Black/Yellow
3	Orange	**8**	Blue	**13**	White/Brown	**18**	White/Violet	**23**	Black/Green
4	Pink	**9**	Violet	**14**	White/Red	**19**	White/Black	**24**	Black/Gray
5	Yellow	**10**	Gray	**15**	White/Orange	**20**	Black/Red	**25**	Black/Pink

DB25BO Screw Terminal Breakout Board

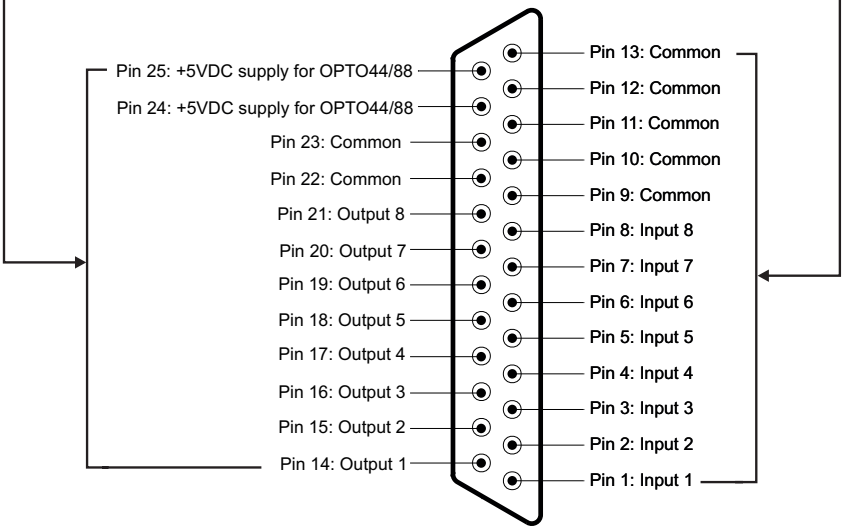
DB25BO



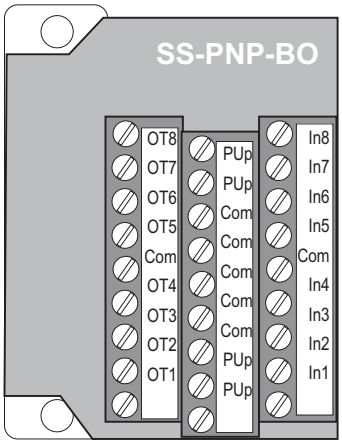
When connected to the SmartStep, the DB25BO terminals will match the pinouts of the 25-pin connector as shown here.

See page 9-5 for Input and Output schematics.

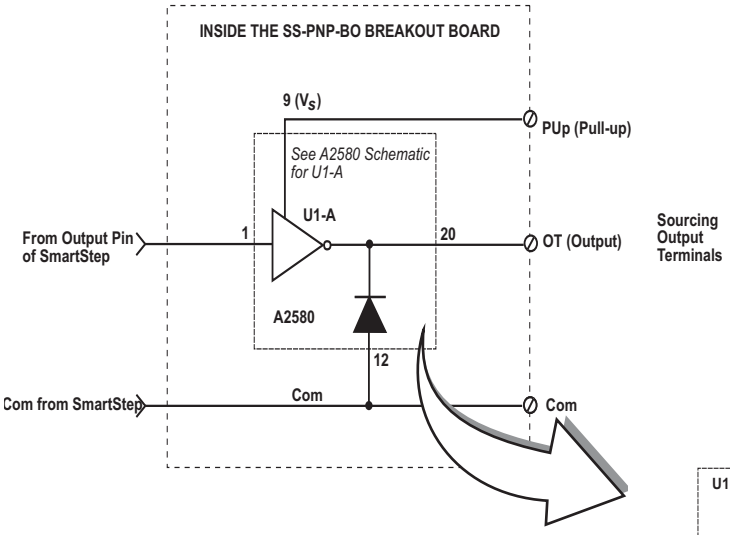
25-pin Connector on SmartStep



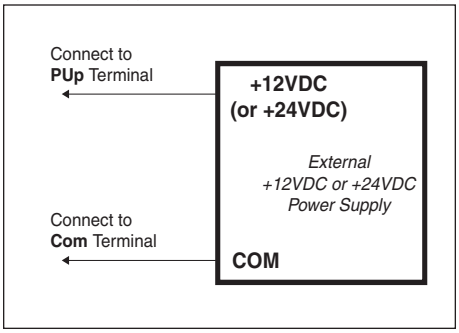
SS-PNP-BO Screw Terminal Breakout Board

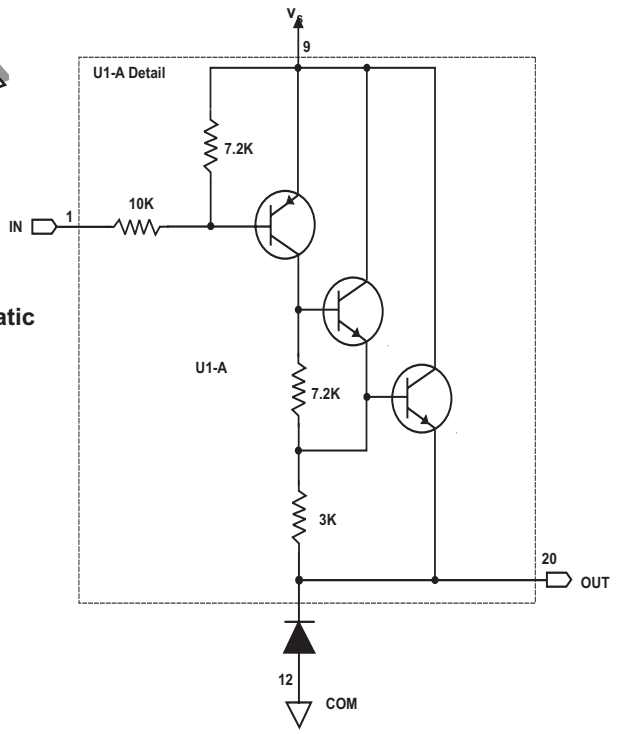


- The SS-PNP-BO Breakout Board converts **NPN** outputs to **PNP** outputs on the SmartStep. The input terminals (In1- In8) are connected directly to the non-converted NPN inputs in the SmartStep.
- Connect I/O to terminals as shown in the illustration on the left.
  In1 - In8 = input (see Input Schematics on page 9-5)
  OT1 - OT8 = PNP outputs
  PUp = pull-up
  Com = common from SmartStep
- An external power supply must be connected as shown in the illustration below right.
- Maximum source current: 100 mA per output.



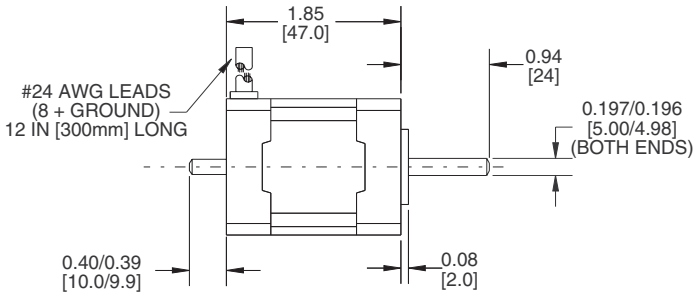
Connecting an External Power Supply

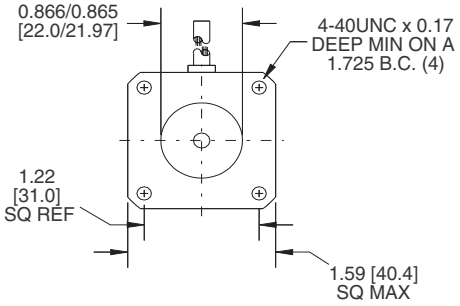




U1-A Detail Schematic

S12 Hybrid Step Motor Specifications

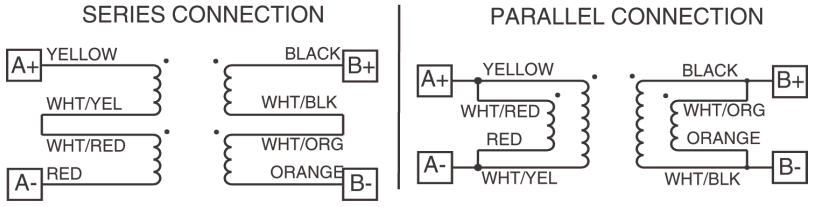




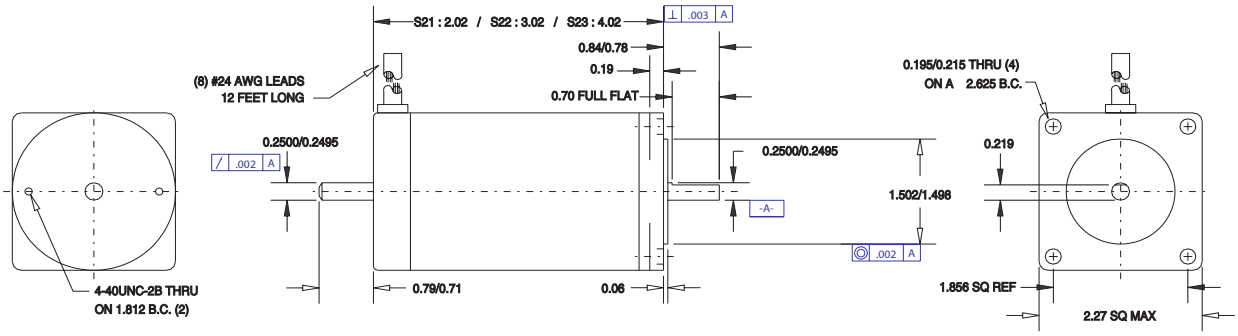
		T=Series	V=Parallel
Electrical Specs.		**S12T**	**S12V**
Continuous Stall Torque	oz-in [N-m]	35 [0.25]	
Recommended Current/Phase	Amps	1.0	2.0
Winding Resistance @ Ambient	Ohms	5.52	1.38
Inductance	mH	8.8	2.2
Max. Winding Temperature	˚F [˚C]	212 [100]	
Mechanical Specs.		**S12T**	**S12V**
Rotor Inertia	oz-in-s^2 [kg-m^2]	5.1x10^{-4}	
Axial Shaft Load	lbs [N]	10 [45]	
Radial Shaft Load @ 0.5 inches	lbs [N]	5 [22]	
Motor Weight	lbs [kg]	0.66 [0.3]	
Step Angle (full step)	degrees	1.8	

Notes:
- Parallel (V) wiring: 50% duty cycle max. above 5 rps (300 rpm).
- IDC step motors require a torque safety margin of at least 30%.

IDC Motor Wiring



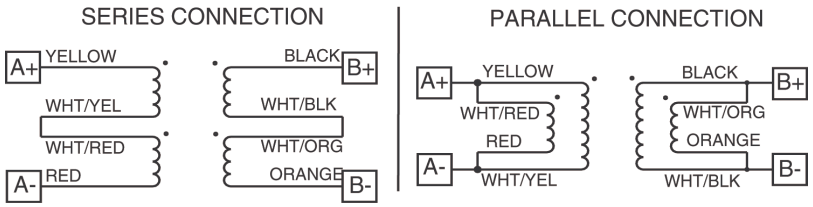
S21/S22/S23 Hybrid Step Motor Specifications



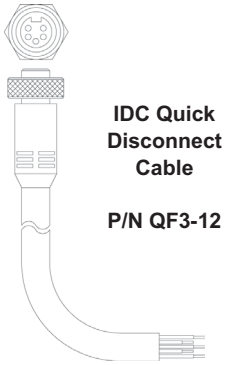
T=Series V=Parallel

Electrical Specs.		S21T	S21V	S22T	S22V	S23T	S23V
Continuous Stall Torque	oz-in [N-m]	65 [0.46]		100 [0.71]		125 [0.88]	
Recommended Current/Phase	Amps	1.2	2.4	1.5	3.0	1.75	3.5
Winding Resistance @ Ambient	Ohms	5.4	1.35	4.8	1.2	4.4	1.1
Inductance	mH	18	4.5	18	4.5	18	4.5
Max. Winding Temperature	˚F [˚C]	212 [100]		212 [100]		212 [100]	
Mechanical Specs.		**S21T**	**S21V**	**S22T**	**S22V**	**S23T**	**S23V**
Rotor Inertia	oz-in-s^2 [kg-m^2]	1.66x10^{-3}[1.17x10^{-5}]		3.31x10^{-3}[2.34x10^{-5}]		4.97x10^{-3}[3.51x10^{-5}]	
Axial Shaft Load	lbs [N]	25 [111]		25 [111]		25 [111]	
Radial Shaft Load @ 0.5 inches	lbs [N]	5.6 [25]		5.6 [25]		5.6 [25]	
Motor Weight	lbs [kg]	1.6 [0.73]		2.4 [1.1]		3.2 [1.5]	
Step Angle (full step)	degrees	1.8		1.8		1.8	

IDC Motor Wiring



Notes:
- Parallel (V) wiring: 50% duty cycle max. above 5 rps (300 rpm).
- IDC step motors require a torque safety margin of at least 30%.

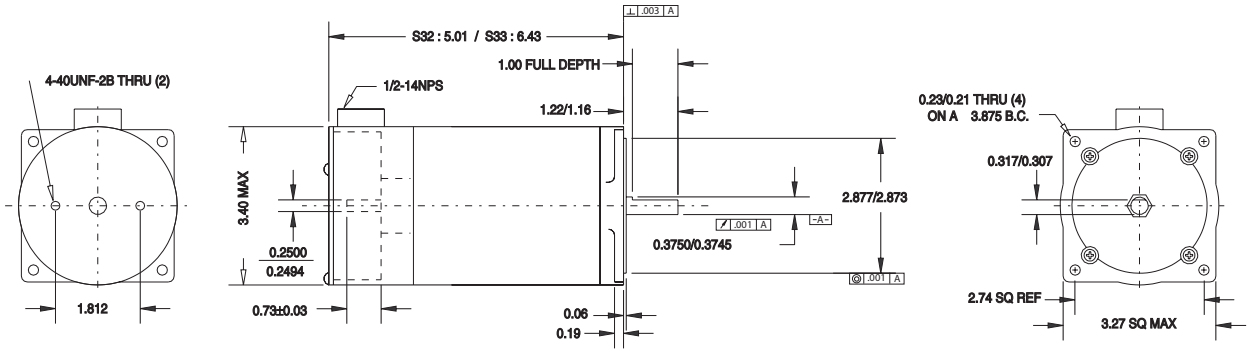


IDC Quick Disconnect Cable

P/N QF3-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

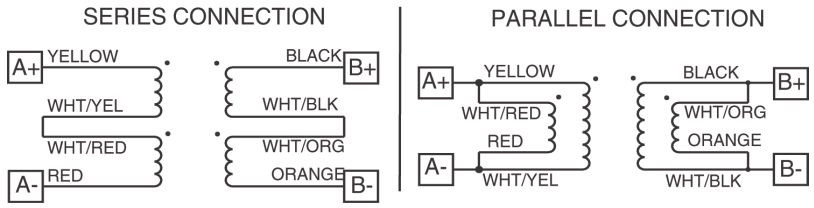
S32/S33 Hybrid Step Motor Specifications

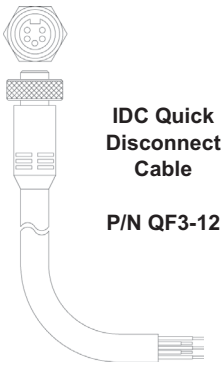


		T=Series		V=Parallel	
Electrical Specs.		**S32T**	**S32V**	**S33T**	**S33V**
Continuous Stall Torque	oz-in [N-m]	300 [7.1]		400 [5.3]	
Recommended Current/Phase	Amps	2.8	5.6	3.5	7.0
Winding Resistance @ Ambient	Ohms	1.03	.26	.96	.24
Inductance	mH	10	2.5	10	2.5
Max. Winding Temperature	˚F [˚C]	212 [100]		212 [100]	
Mechanical Specs.		**S32T**	**S32V**	**S33T**	**S33V**
Rotor Inertia	oz-in-s^2 [kg-m^2]	0.017 [3.51x10^{-5}]		0.0265 [3.51x10^{-5}]	
Axial Shaft Load	lbs [N]	50 [222]		50 [222]	
Radial Shaft Load @ 0.5 inches	lbs [N]	14.5 [64.4]		14.5 [64.4]	
Motor Weight	lbs [kg]	5.1 [2.3]		8.3 [3.8]	
Step Angle (full step)	degrees	1.8		1.8	

Notes:
- Parallel (V) wiring: 50% duty cycle max. above 5 rps (300 rpm).
- IDC step motors require a torque safety margin of at least 30%.

IDC Motor Wiring



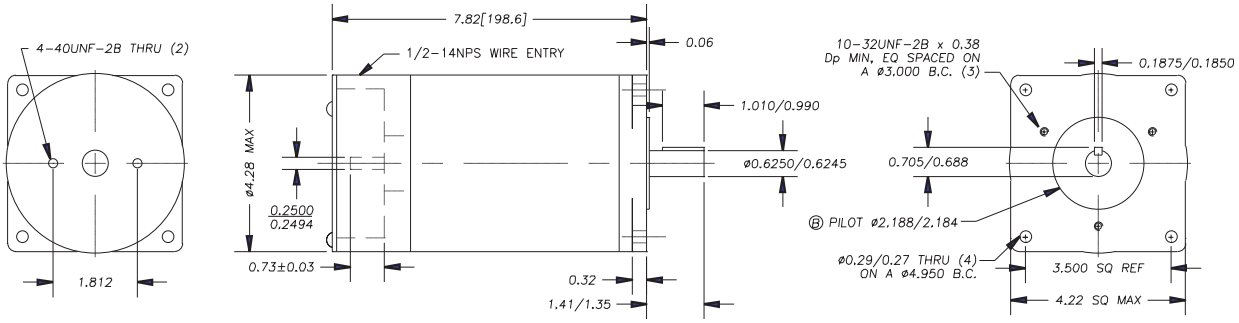


IDC Quick Disconnect Cable

P/N QF3-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

S42 Hybrid Step Motor Specifications

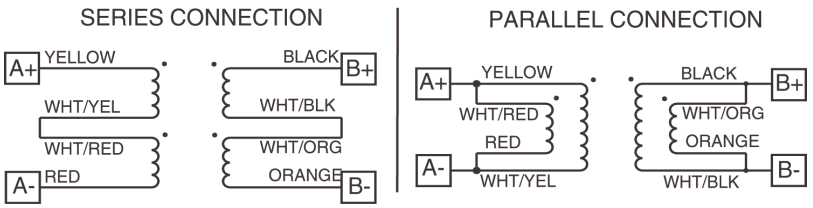


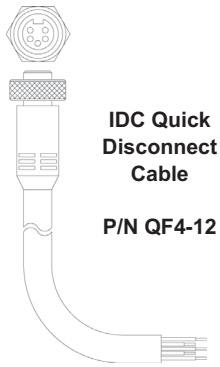
T=Series V=Parallel

Electrical Specs.		S42T	S42V
Continuous Stall Torque	oz-in [N-m]	1000 [7.1]	725 [5.1]
Recommended Current/Phase	Amps	6.0	7.9
Winding Resistance @ Ambient	Ohms	.36	.09
Inductance	mH	7	1.75
Max. Winding Temperature	˚F [˚C]	212 [100]	
Mechanical Specs.		**S42T**	**S42V**
Rotor Inertia	oz-in-s^2 [kg-m^2]	114x10^{-3}[80.5x10^{-5}]	
Axial Shaft Load	lbs [N]	65 [289]	
Radial Shaft Load @ 0.5 inches	lbs [N]	23.6 [105]	
Motor Weight	lbs [kg]	19.1 [8.66]	
Step Angle (full step)	degrees	1.8	

Notes:
- Parallel (V) wiring: 50% duty cycle max. above 5 rps (300 rpm).
- IDC step motors require a torque safety margin of at least 30%.

IDC Motor Wiring



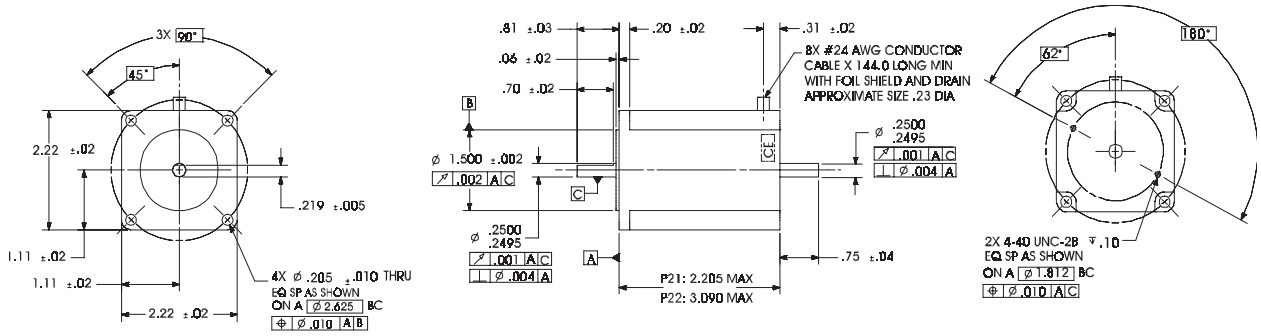


IDC Quick Disconnect Cable

P/N QF4-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

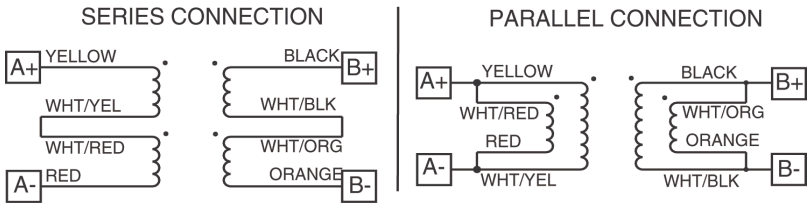
P21/P22 Hybrid Step Motor Specifications



Note: specs. are for both series and parallel wiring unless preceded by T= Series or V= Parallel

Electrical Specs.		P21T	P21V	P22T	P22V
Continuous Stall Torque	*oz-in [N-m]*	100 [0.7]		200 [1.4]	
Recommended Current/Phase	*Amps*	0.7	1.3	1.0	2.0
Winding Resistance @ Ambient	*Ohms*	19	4.7	15	3.7
Inductance	*mH*	79	20	64	16
Max. Winding Temperature	*˚F [˚C]*	248 [120]		248 [120]	
Mechanical Specs.		**P21T**	**P21V**	**P22T**	**P22V**
Rotor Inertia	*oz-in-s^2 [kg-m^2]*	0.0035 [2.48 x 10^{-5}]		0.0061 [4.32 x 10^{-5}]	
Axial Shaft Load	*lbs [kg]*	10 [44]		10 [44]	
Radial Shaft Load @ 0.75 inches [19mm]	*lbs [kg]*	15 [66]		15 [66]	
Motor Weight	*lbs [kg]*	1.9 [0.86]		2.7 [1.23]	
Step Angle (full step)	*degrees*	1.8		1.8	

IDC Motor Wiring



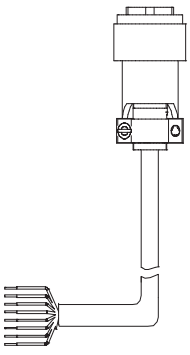
Note:
- IDC step motors require a torque safety margin of at least 30%.

IDC Quick Disconnect Cable

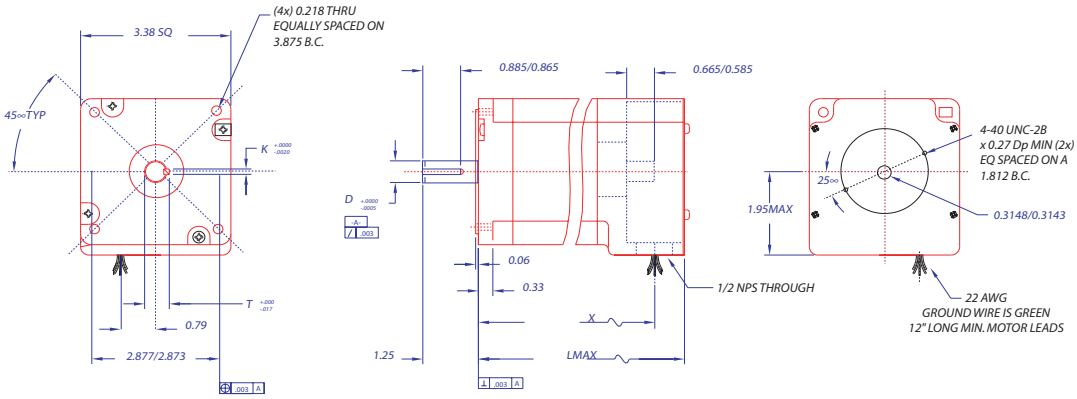
P/N QF4-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

Encoder Color Code & Pin-out for P&K Motors		
Signal	**Pin**	**Color**
A+	B	Red
A-	C	Pink or Purple
B+	N	Green
B-	P	Blue
Z+	M	Yellow
Z-	U	Orange
+5V	K	White
COM	T	Black
Shld	---	---
		Brown (NC)



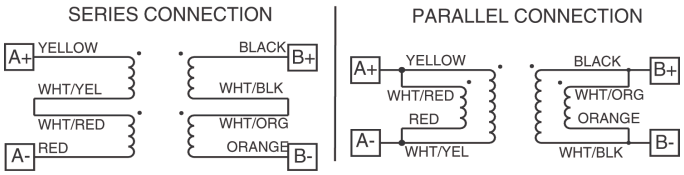
P/K 31, 32, 33 Step Motor Specifications



Motor Dimensions

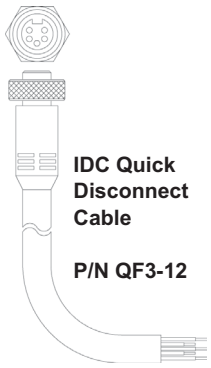
MOTOR	D	K	T	X	LMAX.
P31/K31	.5000	.1250	.555	(3.70)	4.44
P32/K32	.5000	.1250	.555	(5.22)	5.96
P33/K33	.6250	.1875	.705	(6.74)	7.48

IDC Motor Wiring



Note: Specs. are for both series and parallel wiring unless preceded by T=Series or V=Parallel

Electrical Specs.	P31	P32	P33	K31	K32	K33
Continuous Stall Torque oz-in [N-m]	450 [3.2]	920 [6.5]	1260 [8.9]	580 [4.1]	1200 [8.5]	1600 [11.3]
Recommended Current/Phase Amps	T=1.5 V=2.9	T=1.6 V=3.3	T=2.0 V=4.0	T=1.5 V=2.9	T=1.6 V=3.3	T=1.7 V=3.3
Inductance mH	T=56 V=14	T=120 V=30	T=100 V=25	T=56 V=14	T=120 V=30	T=117 V=30
Max. Winding Temperature ˚F [˚C]	212 [100]	212 [100]	212 [100]	212 [100]	212 [100]	212 [100]
Mechanical Specs.	**P31**	**P32**	**P33**	**K31**	**K32**	**K33**
Rotor Inertia oz-in-s^2 [kg-m^2]	.0202	.038	.0567	.0202	.038	.0567
Axial Shaft Load lbs [N]	305 [1350]	305 [1350]	305 [1350]	305 [1350]	305 [1350]	305 [1350]
Radial Shaft Load @ 0.5 inches lbs [N]	65 [285]	65 [285]	110 [489]	65 [285]	65 [285]	110 [489]
Motor Weight lbs [kg]	5	8.4	11.9	5	8.4	11.9
Step Angle (full step) degrees	1.8	1.8	1.8	1.8	1.8	1.8

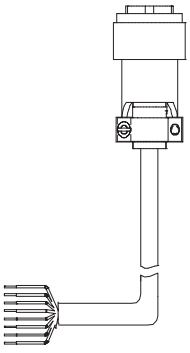


IDC Quick Disconnect Cable

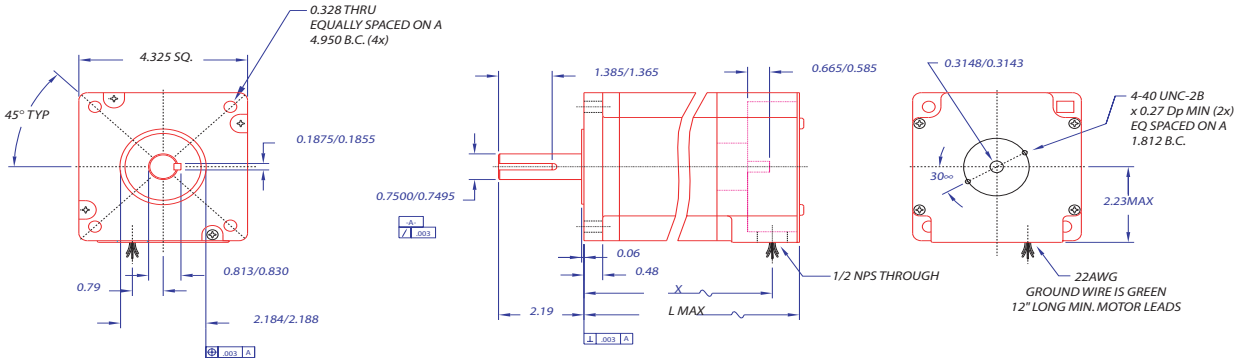
P/N QF3-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

Encoder Color Code & Pin-out for P&K Motors		
Signal	Pin	Color
A+	B	Red
A-	C	Pink or Purple
B+	N	Green
B-	P	Blue
Z+	M	Yellow
Z-	U	Orange
+5V	K	White
COM	T	Black
Shld	---	---
		Brown (NC)



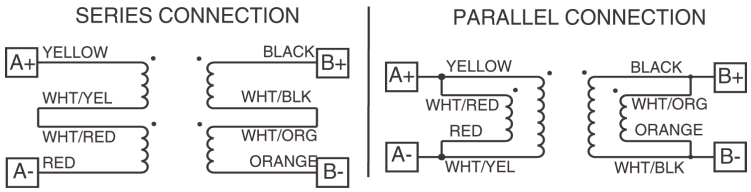
P/K 41, 42, 43 Step Motor Specifications



Motor Dimensions

MOTOR	X	LMAX.
P41/K41	4.46	5.20
P42/K42	6.48	7.22
P43/K43	8.49	9.23

IDC Motor Wiring



Note: Specs. are for both series and parallel wiring unless preceded by T=(series) or V=(parallel)

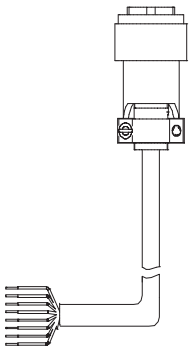
Electrical Specs.	P41	P42	P43	K41	K42	K43
Continuous Stall Torque oz-in [N-m]	1250 [8.8]	2300 [16.2]	3250 [22.9]	TBD	3000 [21.2]	TBD
Recommended Current/Phase Amps	T=2.8 V=5.7	T=3.3 V=6.6	T=3.3 V=6.6	T=2.8 V=5.7	T=3.2 V=6.4	T=3.3 V=6.6
Inductance mH	T=60 V=15	T=84 V=21	T=112 V=28	T=60 V=15	T=60 V=15	T=112 V=28
Max. Winding Temperature ˚F [˚C]	212 [100]	212 [100]	212 [100]	212	212 [100]	212
Mechanical Specs.	**P41**	**P42**	**P43**	**K41**	**K42**	**K43**
Rotor Inertia oz-in-sec	.0783	.1546	.2293	.0783	.1546	.2293
Axial Shaft Load lbs [N]	404 [1790]	404 [1790]	404 [1790]	404	404 [1790]	404
Radial Shaft Load @ 0.5 inches lbs [N]	125 [550]	110 [489]	110 [489]	125	110 [489]	110
Motor Weight lbs [kg]	11	18.4	25.7	11	18.7	25.7
Step Angle (full step) degrees	1.8	1.8	1.8	1.8	1.8	1.8

IDC Quick Disconnect Cable

P/N QF4-12

Wire Color on Quick Disconnect Cables	Drive Connection
Red w/ Black	B-
Red w/ White	B+
Green	*GND
Red w/ Yellow	A-
Red	A+
*Gray-colored Quick Disconnect Cables are shielded - connect shield to GND.	

Encoder Color Code & Pin-out for P&K Motors		
Signal	**Pin**	**Color**
A+	B	Red
A-	C	Pink or Purple
B+	N	Green
B-	P	Blue
Z+	M	Yellow
Z-	U	Orange
+5V	K	White
COM	T	Black
Shld	---	---
		Brown (NC)



Non-IDC Motors

IDC's S Series motors have custom windings to provide optimum dynamic performance with the SmartStep. If you use another manufacturer's motor, it should meet the following guidelines:

1. 2 phase, hybrid, permanent magnet step motor.
2. 4, 6, or 8 lead motor.
3. Series or parallel inductance rating between 4-60 mH for SmartStep and SmartStep23, and 8-240 mH for SmartStep-240. Higher inductance motors will not damage the drive, but they will have limited dynamic performance.
4. A minimum high-pot test rating of 500 VDC.

If possible, use the manufacturer's 160 VDC, bipolar current rating. With **4-lead motors**, the manufacturer's (bipolar) current rating translates directly to the SmartStep current setting.

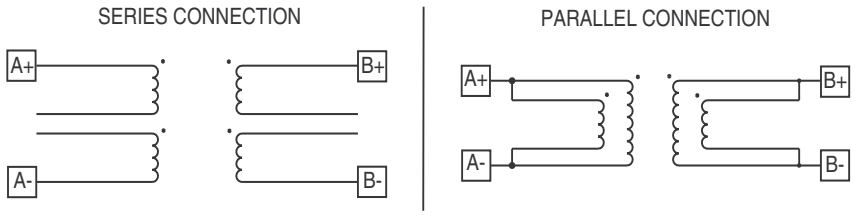
For the proper current setting for **6-lead (unipolar) motors**, use 70% of the manufacturer's rating.

For **8-lead motors**, you have the choice of wiring the motor in series or parallel (see Motor Wiring drawing). In Series, set the current to the manufacturer's bipolar rating. In Parallel, double the bipolar current rating. Care should be taken when running a step motor in parallel to avoid overheating the motor windings. A step motor in parallel may be duty cycle and speed limited. Check with the motor manufacturer for temperature guidelines.

With 4-lead motors, the manufacturer's inductance rating usually translates directly to the SmartStep inductance setting. To determine the 6-lead unipolar motor inductance setting, we normally use 4X the manufacturer's rating. For 8-lead (bi-filar wound) motors in series, set the inductance to 4X the manufacturer's rating. In parallel, use the manufacturer's rating. Again, please check with the motor manufacturer to be sure.

Bi-filar wound, 8-lead, non-IDC motors can be wired in series or parallel as shown in the drawing (though the color code will probably be different). When using a 6-lead unipolar motor, the center taps are left unconnected from the drive and insulated from each other and ground.

Motor Wiring



Calculating $AR_{unloaded}$ for Non-IDC Motors

An Unloaded Anti-Resonance value is required for "Configuring Anti-Resonance" in *Chapter 5, Configuring Your System*.

The following formula is required to calculate Unloaded Anti-Resonance (**$AR_{unloaded}$**):

$AR_{unloaded}$ = 12.987 * LOG[9.3/Vb*SQRT(Tm*Jr))]

Vb = Break Velocity of Knee of Speed-Torque Curve in RPS
Tm = Low Speed Torque of Motor in Nm
Jr = Unloaded Rotor Inertia in Kg-m^2

$AR_{unloaded}$ is also required to calculate your exact Anti-Resonance value (AR).
AR = **$AR_{unloaded}$** -K (see "Configuring Anti-Resonance" in *Chapter 5, Configuring Your System*)

Note: $AR_{unloaded}$ will be different for series and parallel motors, because the speed-torque curve is a component of Vb and Tm.

Chapter 10 - SmartStep Troubleshooting

Symptom	Possible Causes	Possible Remedies
Motor moves the wrong distance	Wrong Gear Ratio.	Check distance units
	Motor stalled.	Check motor current, inductance, anti-resonance settings
		Check Speed Torque requirements of move, lower acceleration.
Motor stalls	Acceleration and/or velocity are too high.	Reduce acceleration and/or velocity.
	Motor configured incorrectly.	Check motor current, inductance, anti-resonance settings.
Motor moves the wrong direction	The motor phases are miswired.	Check per manual, or swap A+ with A-.
	The system's direction is intuitively opposite to yours.	Change the control's direction parameter.
The controller does not respond to keypad input.	The keypad has been disabled.	Check the dip switch settings on the back of the keypad.
The keypad is blank, but the backlight is on.	You have an older keypad that requires new firmware. The SmartStep requires a keypad with Version 2.10 firmware or higher.	Call IDC for a free firmware upgrade to your keypad.
	The DF parameter is set to DF0,0,0,0	See page 5-38 or 8-27.
The keypad is blank and the backlight is off.	The keypad is not receiving a good +5VDC signal.	Check wiring, verify that the +5VDC is between 4.8 and 5.2V.
It is difficult to read the display	The Contrast needs adjustment	Adjust the contrast with the pot on the back of the keypad.
The motor seems to "whine"	The Inductance or anti-resonance setting likely requires adjustment.	Change/Confirm the Inductance setting. The SmartStep operates best with motors 4 mH or above. If this does not help try the anti-resonance setting.

Symptom	Possible Causes	Possible Remedies
The SmartStep ON LED is yellow.	FLASH fault. FLASH has been corrupted.	The operating system and user programs must be reloaded with application developer.
The SmartStep ON LED is red.	A Fault has occurred.	The specific fault can be diagnosed by plugging in a keypad or using serial status commands (SS, SA, SD).
Can't Communicate over RS232, but keypad works.	If the keypad works the SmartStep RS232 port is working. Something else is configured wrong (wiring, configuration, address).	Refer to Chapter 5, RS232 Troubleshooting for in depth help.
Can't Communicate over RS232, and keypad does NOT work.	The keypad is disabled.	Check the label to make sure you don't really have an RS485 version. Either connect to RS485 or call IDC (or your distributor) to exchange units.
		Enable keypad through dip switch on back of keypad and try again.
	The serial port is not working.	Call IDC (or your distributor) for application assistance.
"Hit A Limit"	An EOT+/- switch has been activated.	Either the motion commanded was not correct, or the EOT switch is incorrectly positioned on your system.
"Amplifier Fault"	Multiple drive faults have occurred.	Check the cause of the faults RS-232 SS, SA, and SD commands.

Symptom	Possible Causes	Possible Remedies
"Over Temperature Fault"	Internal Fan or Heatsink Tunnel is clogged or restricted.	Remove obstruction, or clean tunnel by removing unit, use screwdriver to prevent the fan from turning, and blow shop air through the tunnel. Return unit to installation.
	Ambient air in cabinet is too hot.	The SmartStep can produce significant heat. If multiple units are installed next to each other, the cabinet must be adequately ventilated to remove heat.
	The fan is not turning.	With power off confirm the internal fan connection. If connected the fan bearings have probably worn. Contact IDC to obtain a replacement fan.
"Over Current Fault"	The motor is mis-wired or internally shorted. The SmartStep is protected against such shorts, but you will need to correct the problem to clear the fault.	With power off recheck connections. Check SmartStep current setting to make sure you did not accidentally damage a winding by using more current than the motor is rated for. Check to see the motor phases are not shorted to one another or to the case of the motor. The resistance in each phase should be about the same and only a few ohms. If the phases are open or have large resistances the motor is probably damaged and should be replaced.
"Over Voltage Fault"	Too high a bus voltage is present. This is usually caused by a regeneration event that overwhelms our internal power dump circuit. It can also be caused by high line voltage, or voltage spikes.	Eliminate the regen event by reducing the load or make the move less aggressively by reducing the commanded acceleration or velocity. Check your AC line voltage to verify it is within the SmartSteps's limits.
"Interlock Fault"	Motor connector does not contain an interlock wire connection, or motor has been disconnected.	Connect motor connector with Interlock.

Symptom	Possible Causes	Possible Remedies
“Following Error”	Motor stalled.	Confirm proper motor configuration (current, AR, mH). Make a less aggressive move.
	Wrong encoder resolution set.	Setting the encoder resolution incorrectly will cause a following error to occur. Confirm the settings are correct.
“Encoder Wired Backwards”	Encoder position is moving opposite of commanded position.	Check motor and encoder wiring. Reverse phases of either motor or encoder. Consult IDC if unsure.
“Encoder Fault”	Attempted motion in a closed loop mode, and encoder position remained unchanged.	Check encoder wiring, and if the encoder is connected and powered properly.
“Error Finding Home”	Both EOT switches were activated without finding a home switch.	Check if home switch is connected and that the home switch is properly configured as NORM OPEN or NORM-CLOSED.
“Invalid Program”	Attempted to access an empty program (i.e. GT, GS).	Verify program number, or define program called.
“Program Too Large”	Program exceeds 1024 bytes in length.	Split program into smaller programs or reduce program size.
“Insufficient Memory”	All stored user programs exceed 60K.	Reduce program size, or delete programs.
“Invalid Program #”	Program number value exceeds 400, or program name does not exist.	Verify program name and number.
“Unknown Command”	A command not in the IDeal programming set has been issued.	Check program for data entry errors.
“Command Is Too Long”	Command and parameter string exceeds 80 characters.	Reduce command string size.
“Too Many Parameters”	Parameter list exceeds amount supported by command.	Reduce parameter list size.
“Invalid Parameter”	Parameter type is invalid with command.	Verify parameter with command syntax.

Symptom	Possible Causes	Possible Remedies
“Bad Command Syntax”	Command and parameter list has invalid syntax.	Check program for data entry errors.
“Too Many Nested LPs”	Program exceeds 16 nested loops.	Reduce nested loops.
“Too Many Nested GSs”	Program exceeds 16 nested gosubs.	Reduce nested gosubs.
“Too Many Nested EBs”	Program exceeds 16 nested IF blocks.	Reduce number of nested IF blocks.
“Bad Variable Name”	A variable used as a command parameter is undefined or misspelled.	Verify variable name, or define variable with an initial value.
“No Free Variables”	Attempted to define more than 100 user variables.	Reduce number of user variables.
“B8961/2 Command Only”	Command is available on servo SmartDrives only.	Delete the command or purchase a B8961/2.

Product Support

Factory Authorized Distributors

IDC has 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 IDC's systems, as well as other complementary equipment. Contact us at (800) 227-1066 or (603) 893-0588 (from outside the U.S.) for the name of the distributor in your area.

Regional Offices

IDC Distributors are supported by local, direct Danaher Motion Field Sales Engineers (FSE's). Danaher Motion FSE's are available to assist with unusually demanding applications, present on site customer seminars, determine custom product needs, or respond to high volume requirements.

Toll Free Technical Support

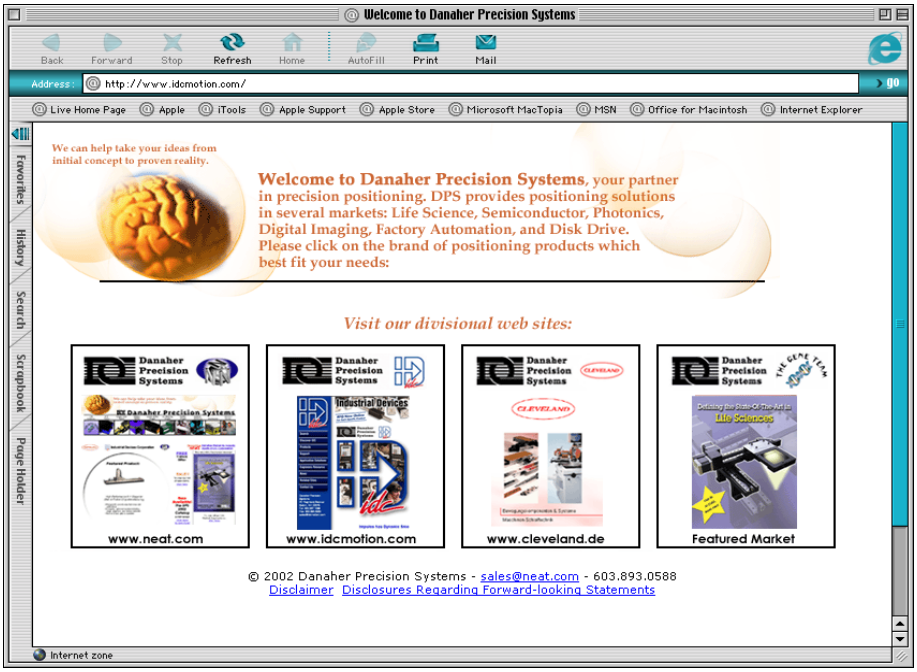
IDC employs a full staff of Applications Engineers, whose full time responsibility is to help you select the proper system, install it correctly, and get it up and operating to your satisfaction. The toll free number is (800) 227-1066. Outside of the United States call (603) 893-0588. Email should be directed to: sales@idcmotion.com.

CAD Library

Actuator, motor, and gear-motor CAD drawings (in .dxf format) are available to save you valuable design time and prevent transcription errors. Low volume requests are complimentary. Check the web site for CAD drawings that may be downloaded.

Web Site

Check us out at **www.idcmotion.com** for information on our products and support.



Warranty & Repairs

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 Rohnert Park, 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 (1-800-277-1066) 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 the address below (be sure to include your RMA # on the shipping label):

Attn: Repair Department
RMA# _ __ _ _
Dahaner Motion
7C Raymond Avenue
Salem, NH 03079

Appendix A - IDC Electric Actuator Ratios

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

One of the first steps in setting up an IDC SmartStep 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 SmartStep. 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. SmartStep Keypad [EDIT-SETUP-MECH-RATIO Menu]
2. Application Developer software [*Setup, Axis* Menu]
3. Direct RS-232C connection [GR Command]

Using the Keypad to Enter Ratio Information

Press:

EDIT **> SETUP > MECH > RATIO**

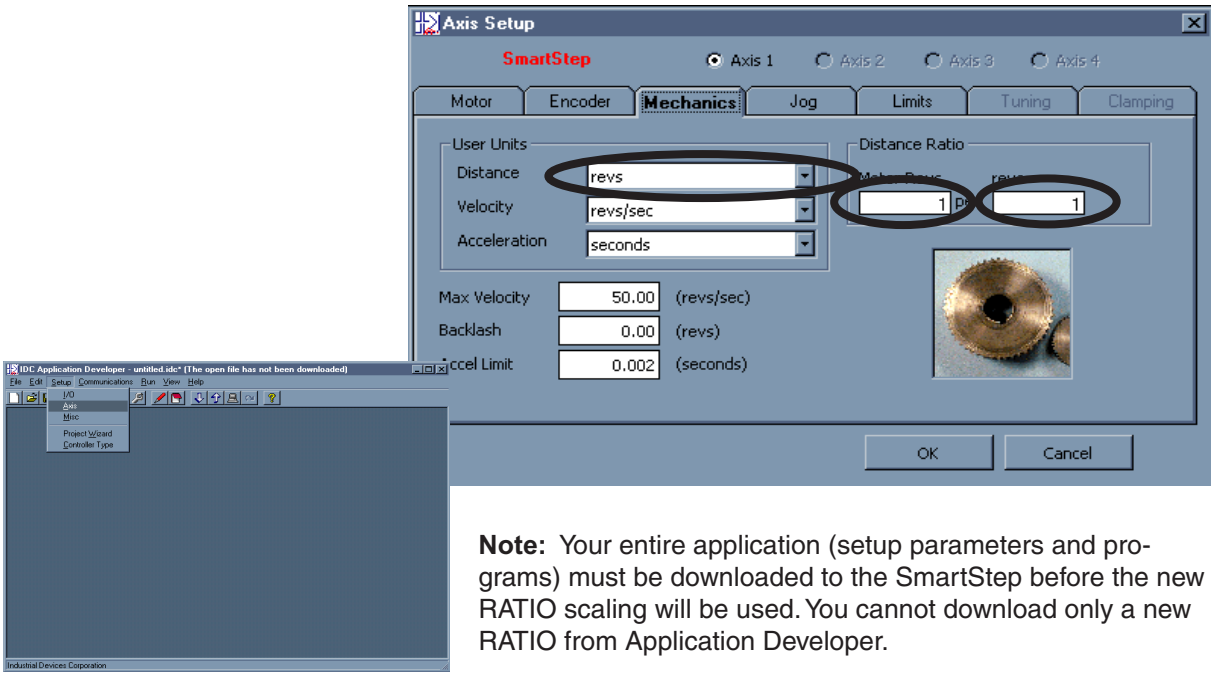
Keypad Display

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

Default: 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

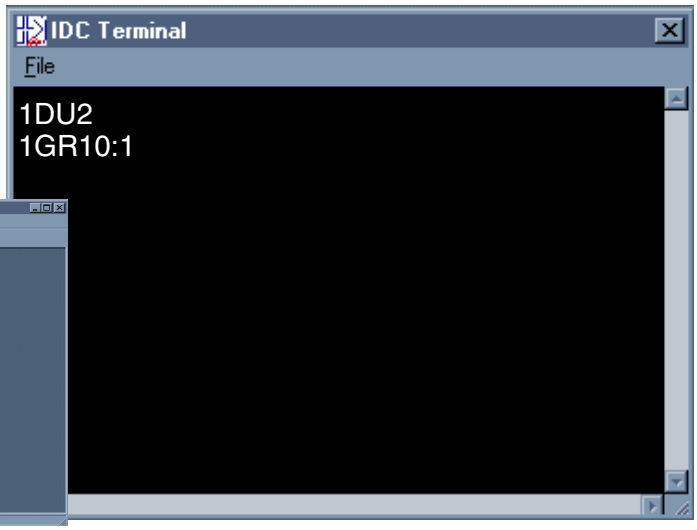


Note: Your entire application (setup parameters and programs) must be downloaded to the SmartStep before the new RATIO scaling will be used. You cannot download only a new RATIO from Application Developer.

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

N2, EC2, EC3, EC4, EC5 Series

N2 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
N2-10-2x	1	2	2	2 to 1	20 to 254
N2-15-2x	1.5	2	3	3 to 1	30 to 254
N2-20-2X	2	2	4	4 to 1	40 to 254
N2-10-5x	1	5	5	5 to 1	50 to 254
N2-15-5x	1.5	5	7.5	75 to 10	750 to 2540
N2-20-5x	2	5	10	10 to 1	100 to 254
N2-25-5x	2.5	5	12.5	125 to 10	1250 to 2540
N2-31-5x	3.125	5	15.625	250 to 16	2500 to 4064
N2-120-5x	12	5	60	60 to 1	600 to 254
N2-10-8x	1	8	8	8 to 1	80 to 254
N2-15-8x	1.5	8	12	12 to 1	120 to 254
N2-20-8x	2	8	16	16 to 1	160 to 254
N2-31-8x	3.125	8	25	25 to 1	250 to 254
N2-120-8x	12	8	96	96 to 1	960 to 254

EC2 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
EC2-10-16B	1	1.5875	1.59	254 to 160	1 to 16
EC2-15-16B	1.471	1.5875	2.33	6350 to 2720	25 to 272
EC2-20-16B	2	1.5875	3.18	508 to 160	2 to 16
EC2-50-16B	5.022	1.5875	7.97	242341 to 30400	9541 to 30400
EC2-100-16B	10.005	1.5875	15.88	183487 to 11552	72239 to 115520
EC2-10-05B	1	5.08	5.08	254 to 50	1 to 5
EC2-15-05B	1.471	5.08	7.47	6350 to 850	25 to 85
EC2-20-05B	2	5.08	10.16	508 to 50	2 to 5
EC2-50-05B	5.022	5.08	25.51	242341 to 9500	9541 to 9500
EC2-100-05B	10.005	5.08	50.83	183487 to 3610	72239 to 36100
EC2-10-04A	1	6.35	6.35	254 to 40	1 to 4
EC2-15-04A	1.471	6.35	9.34	6350 to 680	25 to 68
EC2-20-04A	2	6.35	12.70	508 to 40	2 to 4
EC2-50-04A	5.022	6.35	31.89	242341 to 7600	9541 to 7600
EC2-100-04A	10.005	6.35	63.53	183487 to 2888	72239 to 28880

EC3 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
EC3-10-16B	1.000	1.5875	1.59	254 to 160	1 to 16
EC3-15-16B	1.500	1.5875	2.38	3810 to 1600	15 to 160
EC3-20-16B	2.063	1.5875	3.27	8382 to 2560	33 to 256
EC3-50-16B	5.038	1.5875	8.00	284988 to 35635	112200 to 356352
EC3-70-16B	7.000	1.5875	11.11	327736 to 29491	129030 to 294912
EC3-10-10B	1.000	2.54	2.54	254 to 100	1 to 10
EC3-15-10B	1.500	2.54	3.81	3810 to 1000	15 to 100
EC3-20-10B	2.063	2.54	5.24	8382 to 1600	33 to 160
EC3-50-10B	5.038	2.54	12.80	284988 to 22272	112200 to 222720
EC3-70-10B	7.000	2.54	17.78	327736 to 18432	129030 to 184320
EC3-10-05B	1.000	5.08	5.08	254 to 50	1 to 5
EC3-15-05B	1.500	5.08	7.62	3810 to 500	15 to 50
EC3-20-05B	2.063	5.08	10.48	8382 to 800	33 to 80
EC3-50-05B	5.038	5.08	25.59	284988 to 11136	112200 to 111360
EC3-70-05B	7.000	5.08	35.56	327736 to 9216	129030 to 92160
EC3-10-04A	1.000	6.35	6.35	254 to 40	1 to 4
EC3-15-04A	1.500	6.35	9.53	3810 to 400	15 to 40
EC3-20-04A	2.063	6.35	13.10	8382 to 640	33 to 64
EC3-50-04A	5.038	6.35	31.99	284988 to 8909	112200 to 89088
EC3-70-04A	7.000	6.35	44.45	327736 to 7373	129030 to 73728

EC4/5 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
EC5-10-32B	1.000	0.79375	0.79	254 to 320	1 to 32
EC5-15-32B	1.500	0.79375	1.19	3810 to 3200	15 to 320
EC5-20-32B	2.000	0.79375	1.59	508 to 320	2 to 32
EC5-50-32B	5.110	0.79375	4.06	107777 to 26570	42432 to 265696
EC5-100-32B	10.007	0.79375	7.94	174346 to 21949	68640 to 219488
EC4-10-25B	1.000	1.016	1.02	254 to 250	1 to 25
EC4-15-25B	1.500	1.016	1.52	3810 to 2500	15 to 250
EC4-20-25B	2.000	1.016	2.03	508 to 250	2 to 25
EC4-50-25B	5.110	1.016	5.19	107777 to 20757	42432 to 207575
EC4-100-25B	10.007	1.016	10.17	174346 to 17148	68640 to 171475
EC4/5-10-10B	1.000	2.54	2.54	254 to 100	1 to 10
EC4/5-15-10B	1.500	2.54	3.81	3810 to 1000	15 to 100
EC4/5-20-10B	2.000	2.54	5.08	508 to 100	2 to 10
EC4/5-50-10B	5.110	2.54	12.98	107777 to 8303	42432 to 83030
EC4/5-100-10B	10.007	2.54	25.42	174346 to 6859	68640 to 68590

R2A 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
R2A-10T	1	0.33333	0.33333	1 to 3	10 to 762
R2A-15T	1.5	0.33333	0.5	1 to 2	10 to 508
R2A-20T	2	0.33333	0.66667	2 to 3	20 to 762
R2A-31T	3.125	0.33333	1.04167	50 to 48	500 to 12192
R2A-35T	3.571	0.33333	1.19048	50 to 42	500 to 10668
R2A-120T	12	0.33333	4	4 to 1	40 to 254
R2A-102	1	2	2	2 to 1	20 to 254
R2A-152	1.5	2	3	3 to 1	30 to 254
R2A-202	2	2	4	4 to 1	40 to 254
R2A-312	3.125	2	6.25	50 to 8	500 to 2032
R2A-352	3.571	2	7.14286	50 to 7	500 to 1778
R2A-1202	12	2	24	24 to 1	240 to 254
R2A-105	1	5	5	5 to 1	50 to 254
R2A-155	1.5	5	7.5	7.5 to 1	75 to 254
R2A-205	2	5	10	10 to 1	100 to 254
R2A-315	3.125	5	15.625	250 to 16	2500 to 4064
R2A-355	3.571	5	17.8571	250 to 14	2500 to 3556
R2A-1205	12	5	60	60 to 1	600 to 254
R2A-108	1	8	8	8 to 1	80 to 254
R2A-158	1.5	8	12	12 to 1	120 to 254
R2A-208	2	8	16	16 to 1	160 to 254
R2A-318	3.125	8	25	25 to 1	250 to 254
R2A-358	3.571	8	28.5714	200 to 7	2000 to 1778
R2A-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

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 x 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

A

B

C

D

E

F

G

H

I

P

Q

R

S

T

U

V

W

Keypad Programming Template

Use this template to write MS (Message to Display) command menus or programs exactly as they will appear on the keypad LCD display. Each character-position on the keypad display is represented by a numbered blank square below. Please feel free to make copies of this page for writing your programs.

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

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

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

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

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

Keypad Programming Template

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

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

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

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

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

Summary and Page Reference of All Commands Used with the SmartStep

IDeal™ Programming Commands

AC	Acceleration	6-2	**FK**	Function Key	6-8	**LU**	Loop Until	6-18	**SQ**	Square Root	6-26
BR	Break	6-2	**GH**	Go Home	6-10	**LW**	Loop While	6-19	**ST**	Stop Move	6-27
DA	Distance Absolute	6-3	**GI**	Go Immediate	6-11	**MC**	Move Continuous	6-20	**TD**	Time Delay	6-27
DC	Distance to Change	6-4	**GO**	Start Move	6-13	**MS**	Message	6-22	**VE**	Velocity	6-28
DE	Deceleration	6-6	**GS**	Go to Subroutine	6-14	**ON**	On Condition	6-23	**WT**	Wait	6-29
DI	Distance Incremental	6-6	**GT**	Go to Program	6-14	**OT**	Outputs ON/OFF	6-24			
EA	Enable Amplifier	6-6	**IF**	If Conditional	6-15	**_**	Quote	6-24			
EB	End Block	6-7	**IV**	Input Variable	6-16	**RG**	Registration	6-25			
EN	End Routine	6-7	**LP**	Loop	6-17	**SP**	Set Position	6-26			

Serial Setup Commands

AM	Acceleration Max.	8-27	**FL**	Fault Line Polarity	8-28	**JE**	Jog Enable	8-28	**OP**	OPTO Configuration	8-29
AU	Acceleration Units	8-27	**GR**	Units Ratio	8-28	**JH**	Jog High Velocity	8-29	**PG**	Pos. Maint. Gain	8-29
AR	Anti-Resonance	8-27	**HE**	Home Edge	8-28	**JL**	Jog Low Velocity	8-29	**PU**	Power-Up Program	8-29
DF	Display Format	8-27	**HF**	Home Final Direction	8-28	**MD**	Motor Dir. Reference	8-29	**PV**	Pos. Maint. Max. Vel.	8-29
DU	Distance Unit Label	8-27	**HM**	Homing Mode	8-28	**MH**	Motor Inductance	8-29	**PW**	Password	8-30
DY	Scan Delay	8-27	**HO**	Home Offset	8-28	**MI**	Motor Current	8-29	**RE**	Rest	8-30
EL	Enable Line Polarity	8-27	**HS**	Home Switch	8-28	**MR**	Motor Resolution	8-29	**SN**	Scan Conditions	8-30
EM	Encoder Mode	8-27	**ID**	Input Definition	8-28	**MT**	Motor Type	8-29	**SR**	Stop Decel. Rate	8-30
ER	Encoder Resolution	8-28	**IL**	Idle	8-28	**MV**	Maximum Velocity	8-29	**UN**	Unit Number	8-30
ET	E-O-T Switch Polarity	8-28	**IR**	Pos. Maint. Deadband	8-28	**OD**	Output Definition	8-29	**VU**	Velocity Units	8-30
FE	Following Error Limit	8-28	**JA**	Jog Acceleration	8-28	**OE**	Output State on Event	8-29	**WA**	Waveform	8-30

Serial Immediate Status Commands

CB	Clear Comnd. Buffer	8-35	**MN**	Model Number	8-35	**RS**	Reset System	8-36	**SD**	Tell Drive Status	8-38
IS	Tell Input States	8-35	**OS**	Tell Output States	8-35	**S**	Stop	8-36	**SS**	Tell System Status	8-39
K	Kill	8-35	**PA**	Tell Absolute Position	8-35	**SA**	Tell Axis Status	8-37			

Serial Supervisory Commands

AA	Auto Address	8-40	**EP**	End Prog. Definition	8-40	**OC**	Orig. Configuration	8-41	**UA**	Upload All	8-41
DP	Delete Program	8-40	**EX**	Ends UA or UL	8-41	**PR**	Define Program	8-41	**UL**	Upload Program	8-41
DR	Dwnld. Prog. to RAM	8-40	**LA**	Load All	8-41	**RN**	Run Program	8-41			
EC	RS-232 Echo Enable	8-40	**LS**	List Programs	8-41	**SW**	Tell Software Version	8-41			

Operators, Functions, and Expressions - see pages 7-6 thru 7-7

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

Built-In Variables - see pages 7- 3 thru 7- 5

(AI1) thru (AI6)	Analog input 1 - 6	Read Only	**(1TW)**	Scans input 1-4 for BCD digit	Read Only
(AROWREL)	Current status of any arrow key	Read Only	**(2TW)**	Scans input 1-8 for BCD digit	Read Only
(CPOS1)	Commanded position of axis 1	Read Only	**(TIME)**	Elapsed time (ms) since last power-up or reset	Read Only
(EPOS1)	Encoder position of axis 1	Read Only	**(CRCS)**	Val. of setup checksum	Read Only
(POS1)	Current position of axis 1	Read Only	**(CRCP)**	Val. of program checksum	Read Only
(VEL1)	Cmnd. velocity of axis 1	Read Only	**(SA1)**	Value of axis status	Read Only
(EE1) thru (EE50) **(#F1) thru (#F50)**	Non-volatile variables	Read/Ltd. Write	**(SD1)**	Value of drive status	Read Only
(FKEY)	Val. of Func. Key pressed	Read Only	**(SS)**	Value of System Status	Read Only
(LASTKEY)	Val. of last function key pressed	Read/Write	**(INT98CTRL)**	Enable/Disable (ARM INT98)	Read/Write
(TERM)	Send variable out RS232 Port	Write Only	**(ARM INT98)**	Enable/Disable INT98 if (INT98CTRL) is enabled	Read/Write