Vickers®

General Product Support

SMC20 Motion Controller

Contents

Description

The Vickers SMC20 is a stand-alone, two-axis precision motion controller. It has a built-in electrical power supply, and connects to several other components in a typical motion control system:

- A power source.
- A motion actuator for each axis: (any combination of the following)
  - a drive amplifier and electric servo motor, or
  - a hydraulic servo valve and cylinder or motor.
  - a hydraulic proportional valve and amplifier with cylinder or motor.
- A position feedback sensor for each axis of motion.

In addition, the SMC20 includes communication ports for connecting it to the Hand-Held User Terminal and printer.

The motion controller occupies an 8″ x 3.5″ x 11.5″ case that can be mounted in any standard NEMA-type cabinet or mounted flat using the optional hinge mounting.

Figure 1.1 shows the front panel and connections.

Figure 1.1 SMC20 Front Panel

Features

The SMC20 motion controller presents several features:

- High performance for electric and hydraulic motion control applications.
- Two-axis event coordination capability linked via programmable digital I/O.
- Power supply, controller, and programmable I/Os in one package.
- Hand-Held Operator Interface
- Dual processor design – 16 bit interface microprocessor and 32 bit digital signal processor.
- Plug-in option boards for position sensors and communications.
- Discrete, programmable digital I/O.
- ±10 V differential or single–ended output.
- 0±24 mA, 0±50 mA, 0±100 mA output command. (optional) (Model code selectable)
- Proportional, integral, and derivative (PID) compensation
- Direction sensitive gains to compensate for unequal area cylinders. Also Dither and Gain Break are available in Hydraulic mode.
- Programmability in standard engineering units.
- Ability to change gains within a profile to compensate for load changes, etc.

Front Panel

The SMC20 front panel provides all the controller's connections to the motion control system. Refer to Figure 1-1 to locate the connections (viewed from top to bottom; left column, center column, right column):

Left Column

- Power input (120 VAC @ 2A)
- Isolated DC supply; 24 Volts, unregulated.
- Power 'On' LED indicators.
- Fuse.

Center Column

- Axis 1 (X)
- "Enable" and "CPU" LED indicators.
- Common system. (Opto-isolation power, enable, etc.)
- Axis 2 (Y)

Right Column

- Axis 1 (X) position sensor.
- Axis 2 (Y) position sensor.
- RS232C communication port.
- RS422/485 communication port.

Axis 1 (X) and Axis 2 (Y) connections each include pins for eight inputs and eight outputs. These inputs/outputs are digital.

LED indicators immediately adjacent to Axis 1 and Axis 2 connections show which pins are active.

Communication Interface

Communication between an operator and the SMC20 occurs through the communication interface, a board that plugs into the controller.

The standard communication interface provides two serial ports – RS232C and RS422/485. The RS422/485 port is dedicated to the operator Hand-Held Terminal. The RS232C port is used for a printer.

Hand–Held Terminal

The Hand-Held Terminal, Figure 1-2, connects to the RS422/485 port. This menu driven display and keyboard provides a convenient and simple method of programming the SMC20. The terminal is detachable or it can be secured to the SMC20 unit.

Printer Interface

The RS232C port is configured to be used with a serial printer. Printer set-up is described later in this manual. The controller can list, program and set-up parameters to the printer.

Figure 1-2. User Interface Terminal

Specifications

CPU	16MHz 80C188 and 32MHz DSP
Memory:	
System Eprom	128K
User Memory/Flash Eprom	64K
RAM	64K
17 Digital Outputs with default functions:	16 Programmable Outputs
	Optical Isolation
	1 CPU Dedicated Fault Output
	Current Sourcing (50 mA max)
	Active High (‘On’ = High)
	Optional Active Low
17 Digital Inputs with default functions:	16 Programmable Inputs
	1 Dedicated Interlock Input (ENBL)
	Current Sourcing (2.5 mA max)
	10 Vdc min. to 24 Vdc max, Active High
Servo Outputs	± 10 Vdc Single Ended Standard
	+ 10 Vdc Differential Standard
	Optional ±50 mA Current
	Optional ±24 mA (Component change required)
	Optional ± 100 mA (Component change required)
Serial Interface:	
	Ports: RS232C and RS422/485
	Speed: RS232C — 300 to 38.4K Baud
	Note: Hand–Held Terminal operates at 9600 Baud.
	Settings: 8 data bits, 2 stop bits, no parity
LCD Display	4 Lines x 20 Characters
Power Input	115/230 Vac 50 or 60 Hz, 2A
	or 92 to 264 Vac, 47 to 63 Hz
	or Optional +24 Vdc @ 2A (Model Code Selectable)
Optional External Power Supply:	
	Digital I/O +10 to +30 Vdc
Isolated Supply for Digital I/O	+24 Vdc Nominal 0.5A
Environmental Conditions	0º to 50ºC (Operation or storage) 10% to 90% humidity, noncondensing

Installation

Receiving the Package

When you receive the package that contains the SMC20 motion controller, it should contain the following parts:

- The SMC20 two-axis motion controller.
- One 2-pin isolated DC supply connector (green; in place).
- One 6-pin common system connector (green; in place).
- Two 32-pin, screw-type connectors (32-pin, solder-type DIN connectors are available from Vickers).
- One 4-pin mating connector for AC power.
- One SMC20 Two–Axis Position Controller User's Manual.
- One SMC20 Programmer's Reference Manual.
- One Hand-Held Pendant (4 line display).

You need to supply all other parts for the motion control system, including the necessary wire and cables.

Check the package contents to be sure that it contains all parts. If one or more parts appear to be missing, contact your authorized Vickers distributor. Be prepared to give the SMC20 model number and serial number (located on the outside of the unit), and the description of the missing part.

Registering Your Controller/Manual

When you have checked to be sure that the package contains all SMC20 parts, please take a moment to fill out the Product Registration Card. You will find the card just inside the front cover of the User's Manual.

Among other things, the Product Registration Card asks for the serial number on your SMC20. Even if you do not have an SMC20, sending in the registration card will allow us to keep you posted on changes to this manual.

CAUTION

This controller is used in motion control systems which involve operation of linear actuators and/or rotary motion which can cause machine damage or personal injury.

For Safety Reasons:

–Read this manual and understand operation of the controller prior to installation and operation.

–The user must install proper interlocks, enable fault circuits and emergency shutdown circuits in accordance with accepted safe practices.

Mounting

The SMC20 is designed to be mounted on a panel, typically in an industrial NEMA rated enclosure. Use 4, #10 machine screws.

The maximum ambient operating temperature is 50^{o}C. If temperatures higher than 50^{o}C are likely, supply air conditioning to the cabinet.

CAUTION

For proper ventilation, mount the SMC20 vertically – not horizontally. Damage may occur. Allow 4 inches free clearance above and below the controller for proper air flow.

Connections

The SMC20 front panel contains all connections to other components in the motion control system; for example, the power source, feedback sensors, I/O, or a Hand–Held Terminal.

To complete the connections, you may use one of a variety of devices; for example, plug–in screw terminals, solder terminals, or multiple–pin plugs. For both the isolated DC supply and the common system connections, use the plug–in screw–type terminal supplied with your SMC20. For axis connections, you can use either the 32–pin screw–type DIN connection supplied with the controller or a 32–pin solder–type DIN connector (optional).

This section presents the connections:

- Power supply.
- Common system.
- Axis.
- Position feedback sensor.
- Communications module.
- I/O

Figure 2-1. Screw Locations mm (inches)

Power Supply

The SMC20 is available in two AC power input versions. One version operates from standard 115 VAC, single phase, 50 or 60 Hz, fused at 2 amperes. The other operates from 230 VAC, single phase, 50 or 60 Hz, fused at 2 amperes. The SMC20 is also available with one DC power input version. It operates from +24 VDC, ±5% and is fused at 1 ampere.

Power supply input voltages are model code selectable.

Table 2–1 shows the three versions. Figure 2–2 locates the power supply connection on the SMC20 front panel.

Version	Input Voltage	Fuse
SMC20 – 115	90 to 125 VAC	2 amperes
SMC20 – 230	200 to 240 VAC	2 amperes
SMC20 – 24	24 VDC ± 5%	2 amperes

Table 2-1. Three SMC20 Versions

Figure 2-2. Power Supply Connections

CAUTION

DO NOT TURN POWER 'ON' until you have completed all connections to the SMC20.

To connect power, use the power supply screw–type terminal on the upper left corner of the SMC20 front panel. **Recommendation:** Use 14 to 16 AWG wire. Figure 2–3 shows the connection.

Figure 2-3. Power Wiring for the SMC20

NOTE: 'GND' refers to earth ground. It is important for safety reasons and for proper operation of the controller that the external GND connection is made as described.

The two ground wires tie to each other internally. Connect one ground wire to the machine frame, the other to the power company ground.

When you complete the power supply connection, go on to the common system connections.

Common System Connections

Common system connections provide for signals that affect the controller, but are not axis specific. For example, the V+ connection provides a voltage supply for optically isolated DC input/output signals. You can connect V+ to one of two power supplies:

- A 10 to 30 VDC, user provided power supply.
- The 24 VDC from the isolated DC supply on the SMC20.

Table 2–2 lists the common system connections and the signals they carry. Figure 2–4 locates the common system and the isolated DC supply connections on the SMC20 front panel.

Signal	Function
ENBL	Enable optically isolated **input**. Must be 'On' for servo output command voltage or current to operate.
V+ (Two terminals internally connected)	Provides a voltage supply for optically isolated DC I/O. You can connect it to a user–supplied power source (15 to 30 VDC) or to the 24 VDC isolated DC supply on the SMC20. **Note:** The isolated supply means isolated from all other controller grounds.
CPU	CPU fault optically isolated **output**. A HIGH level indicates that one or both processors are not running. Normal state is open circuit.
COM	Common for V+ isolated power supply.

Table 2-2. Common System Connections

Figure 2-4. Common System Connections

WARNING

The "ENBL" signal should be connected to an external emergency stop circuit. Loss of this input (or low input) causes the command outputs to go to zero regardless of motion processing. This input must be energized to allow motion.

Fault Protection

The SMC20 provides both hardware and software fault protection. The hardware includes two fault protections:

- **Enable** – an optically isolated "enable" **input** signal must remain 'On' if it goes 'Off', controller command outputs go to zero.
- **Internal Watchdog** – if the controller's processing units fail, the controller command outputs go to zero.

Wiring the Common System Connections

Recommendation: Use 14 to 20 AWG stranded wire; further, if you insert more than one wire into a screw terminal, twist the wires together and solder them. For the common system connections, insert the wire into the green screw-type terminal that comes with the SMC20. If you intend to operate from the internal isolated DC supply (24 V) on the SMC20, use its green screw–type terminal which also comes with the controller as shown in Figure 2-5.

Figure 2-5. Wiring the Common System Connections Using the Internal Isolated Supply

Figure 2-6. Wiring the Common System Connections with a User Power Supply

Axis Connections

The axis connections provide the link between the SMC20 and the actuator (motor/drive amplifier). Each axis connection has eight input and eight output digital signals, all optically isolated. Figure 2-7 locates the axis connections on the SMC20 front panel.

Figure 2-7. Front Panel Axis Connections

Pin No.	Name	Description/ Function
Z32	DO+	Analog output (differential) ± 10 VDC full scale **Note 1**
Z30	DO–	
Z28	AGND	Analog output (single-ended) ± 10 VDC full scale **Note 1**
Z26	SO	Analog output (single-ended) ± 10 VDC full scale **Note 1**
Z24	AGND	Analog signal ground
Z22	IN1+	Analog input (differential) channel 1. Programmable usage ± 10 VDC full scale. See page ##.
Z20	IN1–	
Z18	AGND	Analog signal ground
Z16	IN2+	Not used
Z14	IN2–	
Z12	AGND	Analog signal ground
Z10	ADO+	No connection. Not available on standard unit.
Z8	ADO	
Z6	Iout+	Current output high (0 ± 50 mA) **Note 1** (Optionsal)
Z4	Iout–	Current output low (0 ± 50 mA) **Note 1** (Optionsal)
Z2	AGND	Analog signal ground

Note 1: A single differential digital-to-analog conversion channel drives all three outputs simultaneously: DO+/DO– = differential, SO = single-ended, and the current output. Thus, software controlled scaling affects all three outputs equally.

Table 2-3A. Axis Connection Pins – Z32 through Z2 Identical for Each Axis

Pin No.	Name	Description/ Function
D32	TQ	Output — drive torque (current) enable
D30	VEL	Output — drive speed enable
D28	MTN	Output — motion in progress
D26	IP	Output — in position
D24	HC	Output — home sequence complete
D22	ERR	Output — error
D20	O	Output — uncommitted
D18	O	Output — uncommitted
D16	DRV	Input — drive amplifier “OK”
D14	JG+	Input — jog direction +
D12	JG–	Input — jog direction –
D10	LIM+	Input — travel limit +
D8	LIM–	Input — travel limit –
D6	HOM	Input — home switch
D4	RUN	Input — run program
D2	1	Input — run home sequence

Table 2-3B. Axis 1 (X) Connection Pins – D32 through D2

Pin No.	Name	Description/ Function
D32	TQ	Output — drive torque (current) enable
D30	VEL	Output — drive speed enable
D28	MTN	Output — motion in progress
D26	IP	Output — in position
D24	HC	Output — home sequence complete
D22	ERR	Output — error
D20	O	Output — uncommitted
D18	O	Output — uncommitted
D16	DRV	Input — drive amplifier “OK”
D14	JG+	Input — jog direction +
D12	JG–	Input — jog direction –
D10	LIM+	Input — travel limit +
D8	LIM–	Input — travel limit –
D6	HOM	Input — home switch
D4	RUN	Input — run program
D2	1	Input — run home sequence

Table 2-3C. Axis 2 (Y) Connection Pins – D32 through D2

Digital Input/Output

All digital input/output (I/O) are optically isolated and require a power supply (either user-supplied power at 10 to 30 VDC or isolated DC supply at 24 VDC from the SMC20). Further, some digital I/O have a default usage, as Table 2-3B and 2-3C show; however, you can program one or more of them to perform general purpose functions (Programming the SMC20 page ## or the Programming Reference Manual).

Using Input and Output Connections

Table 2-3A lists the name and function/description for each analog pin in each axis connection on the SMC20. Table 2-3B and 2-3C lists the name and function/description for each digital pin. This section gives further information about each pin. The information includes a typical connection.

Figure 2-8. Typical Axis Wiring

Analog Input/Output

DO+ and DO–

Together, the voltage output <u>DO+</u> (Z32) and <u>DO–</u> (Z30) signals constitute a differential analog output with a range of ±10 VDC full scale. The output typically drives a servo. Figure 2-9 shows a typical connection.

Figure 2-9. Differential Voltage Output

SO and AGND

Together, the AGND (Z28) and SO (Z26) signals constitute a single-ended analog output with a range of ±10 VDC full scale. It serves as an alternative method of connecting the SMC20 to a drive amplifier, but is more susceptible to noise than the differential connection. Therefore, if you use Z28/Z26, use them only with <u>cable that is shorter than 20 feet.</u>

Use a two-wire shielded cable: <u>Connect the shield to ground only at the SMC20 end. Do not connect the amplifier end of the shield to ground. Some installations may require shield ground at each unit.</u> Figure 2-10 shows a typical connection.

Figure 2-10. Single Ended Voltage Output

IN1+ (Z22) and IN1– (Z20)

The Z22 (IN1+) and Z20 (IN1–) signals constitute analog input Channel 1. This differential analog input can be used as a variable. Please refer to the Programming Reference Manual for use of this input.

Both connections require twisted pair shielded cable. Figure 2-11 shows a typical connection.

Figure 2-11. Analog Input Typical Connection

Z10 and Z8, Z16 and Z14

These inputs are not used.

IOUT+ (Z6) and IOUT– (Z4)

The IOUT+ and IOUT– signals constitute an optional ±50 mA current output for driving a hydraulic servo valve, or other current driven actuators or amplifiers. Figure 2-12 shows a typical connection.

AGND (Z24, Z18, Z12, and Z2) pins provide analog signal ground.

Figure 2-12. IOUT+ (Z6) and IOUT– (Z4) Current Output

CAUTION

The SMC20 controller can be supplied with <u>'Active High'</u> or <u>'Active Low'</u> digital I/O. Determine which version you have (from model no. on label) and make sure that external devices connected to the controller are compatible. Improper operation, damage, or personal injury can occur if digital I/O signals are inadvertently activated.

Digital Inputs:

Figure 2-13. Digital Inputs

An <u>'Active High'</u> input is 'On' when input voltage is > 10 Vdc. An <u>'Active Low'</u> input is 'On' when input voltage is < 8 Vdc.

Digital Outputs:

Figure 2-14. Digital Outputs

An 'Active High' output when 'On', will provide up to 50 mA of sourcing current, voltage depends on isolated supply and load impedance. An 'Active Low' output sinks up to 50 mA when 'On'.

CAUTION
Applying more than 50 mA of current to an active low output can cause damage to the controller. DO NOT APPLY DIRECT VOLTAGE TO OUTPUTS.

Digital Input/Output

The SMC20 provides eight digital inputs (D16 through D2) and eight digital outputs (D32 through D18) per axis, all opto–isolated.

D32 (TQ)

The D32 (TQ – Torque) signal serves as the drive torque enable output. Connected to the drive amplifier, this output (when 'On') allows the amplifier to supply current to the electric motor, which produces torque. For example, you can connect D32 to the drive switch.

(DR-SW) input on J1 on the Vickers BRM4S Servo Drive.

D30 (VEL)

The D30 (VEL – Velocity) signal serves as the drive speed enable input to the drive amplifier. You can connect D30 to the reference switch (REF-SW) input on J1 on the Vickers BRM4S Servo Drive.

D28 (MTN

The D28 (MTN – Motion) output remains 'On' when the system executes any motion. Ordinarily, the D28 signal indicates that motion is in progress, but not necessarily. The profile being executed may not currently involve motion; that is, it may be waiting for Dwell Time to elapse or an input event to occur. Thus, the output may remain 'On' although no motion is in progress at the moment.

D26 (IP)

The D26 (IP – In Position) output goes 'On' when the axis reaches the position "band" specified by the In Position Error parameter. The signal remains 'On' as long as the axis remains within the band.

D24 (HC)

The D24 (HC – Home Complete) output goes 'On' when the system completes the home sequence.

D22 (ERR)

The D24 (ERR – Error) output goes 'On' when any error condition exists.

D20 (O) and D18 (O)

The D20 and D18 outputs are uncommitted outputs; that is, they have no default usage. You can program their usage as needed.

D16 (DRV)

The D16 (DRV – Drive Amplifier OK) signal provides a system safety feature. It indicates that the drive amplifier is "OK." If the drive amplifier fails, the input goes 'Off' (low). If the input goes 'Off' (low), the SMC20 stops motion on the axis, decelerating at the rate set in the current segment.

D14 (JG+) and D12 (JG–)

The D14 (JG+ — Jog+) and D12 (JG– — Jog–) inputs cause the SMC20 to issue an output that in turn causes axis motion in the '+' direction; on D12, in the '–' direction.

You can affect the jog speed by how long you push on the jog button on the control console:

- One push causes the axis to move one count; another push, another count, and so on.
- Push and hold for one second or less, causes the axis to move at 50% of the Jog Velocity you set in the Configure (CFG) mode.
- Push and hold for more than one second allows the axis to move at 100% of the Jog Velocity you set in the Configure (CFG) mode.

Figure 2-15 shows typical wiring for the jog inputs (D14 and D12).

Figure 2-15. D14/D12 Typical Connections

D10 (LIM+) and D8 (LIM–)

The D10 (LIM+ — Limit+) and D8 (LIM– — Limit–) inputs connect the SMC20 to overtravel limit switches. If the SMC20 receives either input, it decelerates motion on the axis to a stop. Jog motion is only then allowed, and only in a direction opposite of the limit.

Figure 2-8 shows a typical limit switch connection.

D6 (HOM)

The D6 (HOM — Home Switch) input, when received, tells the system that it has reached its home position.

D4 (RUN)

The D4 (RUN) input, when received, causes the system to run the currently active profile (that is, the profile selected by the Run mode).

The Run input and the Jog inputs (D14 and D12, described earlier in this chapter) are mutually exclusive; that is, the Run input is valid only when the Jog input is 'Off' (low), and vice versa. If both the Run and the Jog inputs go 'On' – an illegal state – the SMC20 ignores both.

D2 (I)

The D2 (I) input causes the system to run its home sequence.

Position Feedback Sensor Connection

Figure 2-16. Position Feedback Sensor Connections

Incremental Encoder Connections

The SMC20 interfaces with industry-standard incremental encoders. The encoder output, which is a square wave, consists of "A" and "B" channels in quadrature, with a "C" marker pulse channel. The SMC20 accepts as input, differential line receivers that conform to RS485; however, it will also accept single-ended signals.

The position feedback encoder connection also includes jumper-selectable +5 VDC or +15 VDC to power the encoder. See Appendix B for jumper settings.

Figure 2-16 locates the position feedback encoder connections on the SMC20 front panel. Figure 2-17 shows the pin layout in each encoder connection, and Table 2-4 presents the name and description for each pin in the connection. The encoder connection mates with a DB-9 female connector.

Figure 2-17. Feedback Pin Layout, Pin Layout, Cable End

Pin No.	Name	Description/ Function
1	V+	Positive voltage supply for encoder Selectable by jumper for +5 VDC or +15 VDC (See Appendix B).
2	GND	Common for supply Can be jumper selected to provide -15 VDC (See Appendix B).
3	"A" HI	Differential "A" channel input. RS485 compatible.
4	"A" LO	
5	"B" HI	Differential "B" channel input. RS485 compatible.
6	"B" LO	Differential "B" channel input. RS485 compatible.
7	"C" HI	Differential "C" channel input. RS485 compatible.
8	"C" LO	
9	GND	Power supply common.

Table 2-4. Position Feedback Encoder Connections

As already mentioned, the SMC20 encoder connections will accept single–ended inputs. Figure 2-18 shows the recommended single–ended connections.

Figure 2-18. Recommended Single-Ended Connections

SMC20 Resolver Feedback Board

The SMC20 resolver feedback board provides direct interface between the SMC20 and Vickers motors with 4, 6 or 8 pole resolvers. User selectable resolutions of 16, 14, 12 and 10 bits are provided via jumpers (see Table 2-5A). The SMC20 software automatically checks the jumper settings. Table 2-5B shows the maximum RPM and number of the counts per revolution for each resolution setting.

Resolution	Channel 1		Channel 2	
	JP3-7	JP3-8	JP4-13	JP4-14
16 bit	1	1	1	1
14 bit	1	0	1	0
12 bit	0	1	0	1
10 bit	0	0	0	0

0 = Installed 1 = Removed

Table 2-5A. Resolution Jumpers

Resolution	Max RPM			Counts/Mech Rev		
16 bit	500	350	250	131,072	196,608	262,144
14 bit	2,000	1,400	1,000	32,768	49,152	65,536
12 bit	8,500	5,500	4,300	8,192	12,288	16,384
10 bit	10,000	10,000	10,000	2,048	3,072	4,096

Table 2-5B. Max RPM & Counts

Wiring and Jumper Selection

Connections are made through two male DB9 connectors, one for each axis. Figure 2-19 shows the pin assignments as well as jumper locations for resolution selection.

Figure 2-19. Resolver Board Setup

Communications Connections

The standard SMC20 provides two communication connections:

- An RS232C-compatible port for printer.
- An RS422/485-compatible port for Hand-Held Programming Terminal.

Figure 2-20. Communication Connections

For normal operation you only need to plug the Hand-Held Terminal into the RS485 connector. You can use the RS232C port to list programs (programs that have been setup with the Hand–Held Terminal) to a printer or a personal computer. Figure 2-21 below shows the RS232C connection.

Figure 2-21. RS232C Pin Layout, Cable End

Pin No.	Name	Description
1	DCD	Not used
2	RX	Serial data input.
3	TX	Serial data output.
4	DRT	Not used
5	COM	Signal common (ground).
6	DSR	Not used
7	RTS	Not used
8	CTS	Not used
9	+5V	+5V (optional)

Table 2-6. RS232C Port Signals

Installation Checklist

The following checks should be checked prior to applying power to the SMC20 controller.

CAUTION
Use of this or similar controllers involves motion of electric (or hydraulic) actuators which have the potential for exerting force or high velocity motion, which can cause machine damage or personal injury. Verify all connections before applying power to the controller. It is a must that the User install Emergency Shutdown provisions which operate independently of this controller. The Emergency Shutdown should remove power from actuators and should be easy to reach by operators at all times.

1.	Inspect controller for damage or missing parts. If there is any difficulty, contact your supplier.
2.	Be sure the controller is mounted in accordance with mounting and ventilation recommendations. (See page ###.)
3.	Connect AC power. Be sure that the GND connection is made to a suitable earth ground. [Green Wire]
4.	Perform a power up test as follows: – If you have pre–wired the unit, temporarily unplug both 32 pin axis connectors. – Apply AC power and verify that the four (4) power supply monitor lights are on ISOL PWR, +5V, +15V, –15V. These LED's are located on the left.
5.	Turn power 'Off' and re–install the axis connectors. Install or check all wiring required for your application. **Remove electric and/or hydraulic power from the actuators.** Perform power up test as in #4 above.
6.	Inspect enable and safety circuits wiring for correctness. **Remove electric and/or hydraulic power from the actuators.** Perform power up test as in #4 above.
7.	Perform setup procedures in Chapter 3 before applying power to the actuators.

Table 2-7. Checklist

Initial Set-Up

Introduction to Programming

To operate the SMC20 you must have a program (a series of commands that define, monitor, and govern the controller's activities).

Use of the Hand-Held Terminal makes it unnecessary to learn a special language. It is still necessary to understand the order in which various features should be used and to properly interpret the abbreviated instructions made necessary by the small input device display.

The Hand-Held Terminal is attached to the RS422/485 port on the lower right corner of the SMC20 front panel.

Hand–Held Terminal

The Terminal has five distinct modes of operation, three of them for programming, plus the Run and Monitor modes.

Monitor	(MON)
Configuration	(CFG)
Variables	(VAR)
Program Segment	(PRG)
Under Program, there is a major sub-mode, Segment, which does not have a panel button.	
Run	(RUN)

When power is turned on, the screen displays the hardware version numbers and a message: "Hit <Clear Entry> Key" as shown below:

```
SMC20   VER   X   X                  F1
                                     F2
                                     F3
HIT  <CLR ENTRY>  KEY                F4
```

If no security code has been set up, pressing any key will automatically set the screen display to the Monitor (MON) mode. As shipped, the security code is set to a default value of 0, which implies no protection.

```
MONITOR MOTION ↑ AXIS                F1
POS 1    0  0000     <SEL            F2
PER 1    0  0000     <SEL            F3
VEL 1    0  0000   ↓ <SEL            F4
```

Each of the modes, MON, CFG, VAR, PRG, RUN is discussed in detail in the Programming section (page ##).

You can change to different modes by pressing the corresponding key. **MON, CFG, VAR, PRG, RUN**.

When words appear at the right edge of the screen aligned with F1, F2, F3, or F4, the word indicates the function of that blue button for that screen. The functions vary with the screen. When Up/Down arrows appear, pushing the green arrow keys will change the screen or one of the choices on the screen. Continued pushing of either Up or Down keys will go through the entire list of screens or choices available in that mode or sub-mode. The choices always form a circular list.

When screens say EDIT, pushing that key leads to a lower level with several screens. When the screen shows <EDT, a single value is to be edited.

On some screens the word EDIT is spelled out. Selecting this function takes the user into other screens for editing. The <EDT form means the value pointed to (and in brackets) can be edited. No screen changes will occur.

On screens for value or name entry, CLEAR ENTRY, ENTER, and BACKSPACE are active. F1 and F2 can be used to move the cursor if the entry is to be edited rather than being cleared and completely re-entered. To delete a character or a digit, position the cursor (blinking square) to the right of the character to be deleted and use the backspace key. Otherwise, a character entered at the cursor is inserted ahead of the position where the cursor is, and the cursor is pushed to the right.

When values are entered, in addition to pushing ENTER on the value input screen, it is necessary to push SAVE on a preceding screen (after ENTER).

A program is not saved against power turn–off until it is SAVED to FLASH MEMORY from the CFG mode. It is therefore, a good idea to save to FLASH Memory frequently.

The SMC20 Programming Structure

Each axis is independently programmed at all levels. See Figure 3-1.

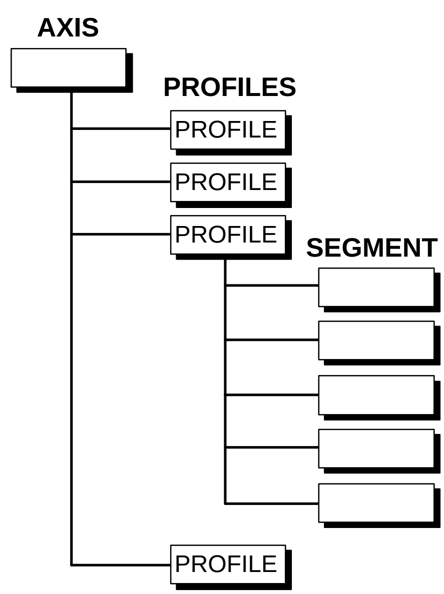

Figure 3-1. The SMC20 Programming Hieracrcy

Axis

One actuator and associated equipment and SMC20 connections.

Profile

A profile is a description of a set of motions and associated input and/or output instructions always ending with the axis returned to zero velocity, but not necessarily the starting position.

In motion control applications, a controlled axis typically executes a series of moves with or without intermediate stops and eventually returns to the starting position. This motion is frequently represented by a graph with position on the X (horizontal) axis and velocity on the Y (vertical) axis.

Segment

Defines the events (I/O) or motion over a portion of a profile which consists of a maximum of one acceleration and one deceleration.

The number of segments in a given profile and the number of different profiles which can be programmed on an axis is not fixed, but instead is governed by the available memory. The SMC20 can store 900 segments. Each profile uses at least one segment but may use many segments.

Figure 3-2A. Segmentation

Configuring Versus Programming

Associated with each of the elements of the SMC20 hierarchy (**AXIS, PROFILE, SEGMENT**) is data or parameters that must be entered.

Some of the data (or parameters) are entered only once for the SMC20.

Typically this type of data applies to the **SYSTEM** or an **AXIS**. Examples of this type of data are Maximum Axis Speed, Jog Speed, Maximum Acceleration, etc. The reason they are only entered once is that these parameters are absolute maximum values or other constants (i.e. gain ratios) that relate to the real machinery hardware not a particular program being run on the SMC20. The data that is only entered once or changed very infrequently is entered in the Configuration mode (CFG).

This data is entered in the configuration mode.

Other types of data are changed frequently. For example, the position that an axis is to move to in a given profile, the speed at which it is to move, whether a limit switch (an input) is to be checked, etc., are all things that depend only on that particular **SEGMENT** or group of **SEGMENTS** (a **PROFILE**). The data that is changed all the time is entered in the Program (PRG) mode.

The type of data and the level that it applies to is shown in Figure 3-2B. Also shown is the mode (CFG or PRG) where the data is entered.

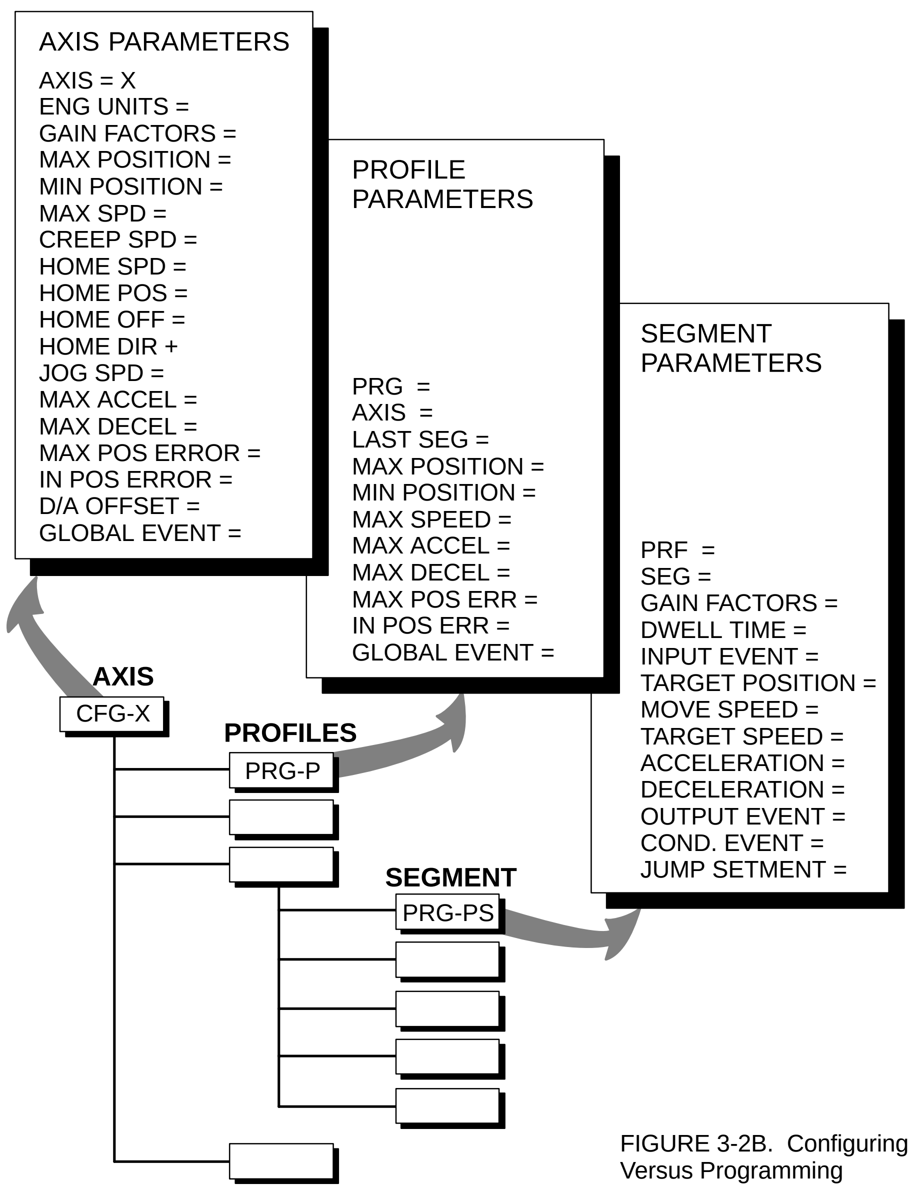

FIGURE 3-2B. Configuring Versus Programming

Set–Up Axis Configurations

Perform the installation and check-list in the Installation section (page ##).

CAUTION
At this point, power should still be removed from electric or hydraulic axis actuators to prevent unintended motion or damage.

Apply power to the controller. Get into Monitor (MON) mode as indicated on page ##. You can change to different modes by pressing the corresponding key. **MON, CFG, VAR, PRG, RUN**.

Monitor Mode

The Monitor mode (MON) is used to monitor motion parameters and input and output status. Three screens are available:

```
MON

MONITOR MOTION ↑ AXIS      F1
POS 1    0  0000     <SEL  F2
PER 1    0  0000     <SEL  F3
VEL 1    0  0000    ↓<SEL  F4

MONITOR I/O       ↑        F1
                     PREV  F2
ADDRESS: [ 0  ]      <EDT  F3
(NO ALARMS)       ↓  NEXT  F4

AXIS[ 1] ALARMS    ↑ AXIS  F1
FLAGS(00000000)            F2
                      ←    F3
(NO ALARMS)        ↓  →    F4
```

Figure 3-3. MON Mode Screens

The first screen allows you to monitor Axis Position, Position Error, and Velocity Command for each axis. The available parameters for viewing are:

Axis 1

POS1 – actual position

PER1 – position error

VEL1 – velocity command

Axis 2

POS2 – actual position

PER2 – position error

VEL2 – velocity command

Note: See pages ##–## for information about other parameters displayed in monitor mode.

Any of the above parameters can be displayed in any of the three fields – F2, F3, F4 by pushing the corresponding **<SEL (F2, F3, F4)**. For example, if the Axis 1 screen is currently displayed (POS1, PER1, VEL1), push **<SEL–F3** once and the following screen will be displayed. Pushing **AXIS-F1** will reset the screen to all Axis 1, or all Axis 2 parameters.

```
MONITOR MOTION ↑ AXIS      F1
POS 1    X XXXX     <SEL   F2
POS 2    X XXXX     <SEL   F3
VEL 1    X XXXX    ↓<SEL   F4
```

Pushing **F1** again will show the following screen:

```
MONITOR MOTION ↑ AXIS      F1
POS 2    X XXXX     <SEL   F2
PER 2    X XXXX     <SEL   F3
VEL 2    X XXXX    ↓<SEL   F4
```

The second screen (reached by Up/Down arrow keys) is used to monitor Alarms (Error Flags). See General Faults section (page ##).

```
AXIS[ 1] ALARMS    ↑ AXIS   F1
FLAGS(00000000)             F2
                            F3
(NO ALARMS)        ↓ NEXT   F4
```

The Monitor I/O screen (reached by Up/Down arrow keys) is used to monitor I/O status.

```
MONITOR I/O       ↑         F1
                     PREV   F2
ADDRESS: [ 0  ]      <EDT   F3
STATE: (0)        ↓  NEXT   F4
```

CFG-System Parameters section (page ##) describes input and output address assignment. You can monitor the state (ON = 1, OFF = 0) of any address by scCFG-rolling through the addresses with the **PREV–F2** or **NEXT–F4** keys. A specific address can be accessed directly by pressing **<EDT–F3** and then entering in the address number, then press ENTER.

A description of parameters is available for viewing on the Monitor screen. In addition to Position, Velocity and Error Parameters, the following are available by scrolling (left–right) each line:

Axis 1 or Axis 2

A/D – Output in analog-to-digital counts. Used for diagnostic purposes.

D/A – Analog input in counts.

BRC – Number times following error exceeded maximum value.

RAW – Feedback counts – for absolute encoders such as resolvers.

P (X) Seg ( ) – Displays program number and segment number currently being executed.

Configuration Mode

Press **CFG** to bring up the Axis Configuration mode. [Note: If the unit has previously been programmed with a security code other than 0, you will have to know the security code to proceed further than Monitor mode.] Press Up/Down arrow keys to change from one screen to the next.

Configuration mode is required to set each axis parameters:

- Axis Gains
- Position Sensor Scale Factor (counts per engineering unit)
- Software Travel Limits
- Home Parameters
- Maximum Acceleration, Deceleration, Move Velocity and other motion parameters
- Input/Output Requirement

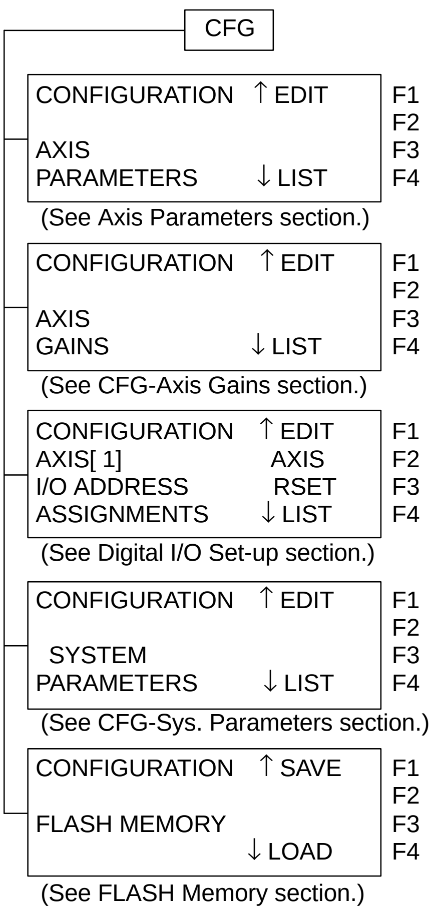

Figure 3-4. CFG Mode Screens

CAUTION

The first time a new controller is powered up (with no previous programming), the axis gains are set to zero to prevent unintended motion. However, analog circuit offset voltages and external devices such as servo amplifiers can have offsets which can cause 'drift' motion. Axis gain should be set to some value before applying power to the actuators.

Before proceeding, you should determine the parameters applicable to your system.

- Position Scale Factor (i.e. counts/inch)
- Maximum Velocity, Acceleration, Deceleration you desire (Equipment should be capable of attaining these values).
- Home Requirements.

The CFG mode is used to set parameters applicable to all programs for an axis. Either axis can be selected. Settings for one axis do not affect the other axis. This mode is shown in five sections. The axis parameters should be set before trying to activate any of the axes.

When the CFG mode switch is pushed, one of the five screens in Figure 3-4 comes up. The others are reached by using the arrow keys.

Any of F1 through F4 not listed for a particular screen is not active with that screen.

The following sections explain in detail how to use the CFG mode screens. To proceed with a quick initial set-up, you only need to enter axis parameters and axis gains (see Quick Set-Up section page ##).

Axis Parameters

In the CFG mode use Up/Down arrows to go to Configuration Axis Parameters screen. Push **F1–EDIT**. Then use the arrow keys to get the screen for a particular parameter.

These parameters apply as DEFAULTS and LIMITS to all axis programs and all segments of those profiles. These values can be **lowered**, but not increased by values entered in the Program (PRG) mode. Therefore, these values should represent the maximum values judged obtainable or judged desirable for this axis.

Note: If these axis parameters are changed after profile parameters or program segments have been entered, the stored segments/profiles WILL NOT be re–checked for out of bounds values and where a profile or segment used default values. These WILL NOT be revised. If it is expected that a particular parameter will be changed, it would be better to use a VARIABLE in each segment where that parameter is used. See Programming section page ##.

```
AXIS[  1]        ↑ QUIT    F1
POSITION           AXIS    F2
ENG UNIT           SAVE    F3
[1.0000    ]     ↓ <EDT    F4
```

```
AXIS[  1]        ↑ QUIT    F1
ENGINEERING        AXIS    F2
UNIT TYPE          SAVE    F3
[EU ]            ↓ <SEL    F4
```

```
AXIS[  1]        ↑ QUIT    F1
LOOP UPDATE        AXIS    F2
RATE               SAVE    F3
[0.1995 ]        ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MINIMUM                AXIS    F2
POSITION (EU)          SAVE    F3
[2147483647.00]      ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MINIMUM                AXIS    F2
POSITION (EU)          SAVE    F3
[-2147483648.0]      ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MAXIMUM POS            AXIS    F2
ERROR (EU)             SAVE    F3
[2000.0000      ]    ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
IN POSITION            AXIS    F2
ERROR (EU)             SAVE    F3
[10.0000       ]     ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MAXIMUM                AXIS    F2
SPEED (EU/SEC)         SAVE    F3
[399999.9388   ]     ↓ <EDT    F4
```

```
AXIS[  1]  (VAL)     ↑ QUIT    F1
FOR JOG SPEED          AXIS    F2
SELECT VARIABLE        <VAR    F3
OR ENTER VALUE       ↓ <VAL    F4
```

```
AXIS[  1]            ↑ QUIT    F1
                               F2
EVENT                  <DEL    F3
CONDITION           ↓ <EDIT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MAXIMUM                AXIS    F2
ACCEL (EU/S^2)         SAVE    F3
[199743.1951   ]     ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
MAXIMUM                AXIS    F2
DECEL (EU/S^2)         SAVE    F3
[199743.1951      ]  ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
D/A OFFSET             AXIS    F2
  (D/A COUNTS)         SAVE    F3
[0.0000   ]          ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
HOME                   AXIS    F2
POSITION (EU)          SAVE    F3
[0.0000   ]          ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
HOME                   AXIS    F2
OFFSET (EU)            SAVE    F3
[0.0000   ]          ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
HOME                   AXIS    F2
SPEED (EU/SEC)         SAVE    F3
[199999.9694   ]     ↓ <EDT    F4
```

```
AXIS[  1]            ↑ QUIT    F1
HOME                   AXIS    F2
DIRECTION              SAVE    F3
[RETURN   ]          ↓ <SEL    F4
```

```
CONFIGURATION   ↑ EDIT    F1
                          F2 Not Used
AXIS                      F3 Not Used
PARAMETERS      ↓ LIST    F4 List to RS232 Printer Port
```

Figure 3-5. CFG Axis Parameters

Axis Parameter Screens

These screens are identical except for the parameter name.

POSITION ENG UNIT Position feedback counts per measurement length. (Depends on Transducer and units used).

LOOP UPDATE RATE Range .2 to 19 ms. This parameter can be useful when tuning an axis for performance. The Default Value is 0.200ms.

MAXIMUM POSITION	(EU)
MINIMUM POSITION	(EU)
MAXIMUM POSITION ERROR	(EU)
IN POSITION ERROR	(EU)
MAXIMUM SPEED	(EU/SEC)
CREEP SPEED	(EU/SEC) Error recovery speed
MAXIMUM ACCELERATION	(EU/SEC^2)
MAXIMUM DECELERATION	(EU/SEC^2)
D/A OFFSET	(D/A COUNTS)
HOME POSITION	(EU)
HOME OFFSET	(EU)
HOME SPEED	(EU/SEC)

For each above,

F1	Quit, Go to previous screen
F2	Axis, Selects 1 or 2
F3	Save, Saves newly entered value to RAM Memory
F4	Brings up the Parameter screen

Push Enter.

```
AXIS[  1]          ↑ QUIT
POSITION             AXIS
ENG UNIT             SAVE
[1.0000    ]       ↓ <EDT
```

F1 QUIT to Previous Screen
F2 Select Axis 1 or 2
F3 SAVE newly entered value to RAM memory
F4 Select engineering unit type

FT = Foot　　MM = Millimeter
RV = Revolution　　CM = Centimeter
EU = Engineering Unit (counts)

```
AXIS[  1]          ↑ QUIT
HOME                 AXIS
DIRECTION            SAVE
[RETURN   ]        ↓ <SEL
```

F1 QUIT to Previous Screen
F2 Select Axis 1 or 2
F3 SAVE newly entered value to RAM memory
F4 Selects RETURN or ADVANCE Direction

```
AXIS[  1]  (VAL)   ↑ QUIT
FOR JOG SPEED        AXIS
SELECT VARIABLE      <VAR
OR ENTER VALUE     ↓ <VAL
```

F1 QUIT
F2 Select 1 or 2
F3
F4

```
AXIS[   ]          ↑ QUIT
JOG SPEED IS         AXIS
ASSIGNED TO          SAVE
VARIABLE #C        ↓ <EDT
```

F1 QUIT to Previous Screen
F2 Select 1 or 2
F3 SAVE to RAM
F4 Enter Variable Number

```
AXIS[  1]          ↑ QUIT
JOG SPEED            AXIS
VALUE (IN/SEC)       SAVE
[            ]     ↓ <EDT
```

F1 QUIT to Previous Screen
F2 Select 1 or 2
F3 SAVE to RAM
F4 Enter JOB Speed

Figure 3-6. Engineering Unit, Home Direction and Job Speed Screens

CFG–Axis Gains

In the CFG mode, use Up/Down arrows to show Configuration Axis Gains screen. Push **Edit–F1**. Use Up/Down arrow keys to obtain any of the screens of Figure 3-7. The function key action is the same for all screens.

- F1 Quit, Go to previous screen
- F2 Axis, Selects 1 (X) or 2 (Y)
- F3 Save, Saves new value to RAM Memory
- F4 <EDT, For all parameters. This brings up one of the Axis Gains screens. For Integrator Delay, when SAVE is pushed, the value entered is adjusted to be an exact multiple of the processor cycle time.

These are ALL AXIS-SPECIFIC gains. For a discussion of how to select the values to use, see Appendix A on PID adjustment. For reference, the gains are listed the following page with brief definitions.

FS	Means full scale controller output
EU	Means engineering units
Kp	Proportional gain
Ki	Integrator Gain
Kd	Derivative Gain (velocity)
Integrator Delay (Sec)	The integrator in the PID compensation in each
Integrator Band (EU)	axis of this controller is a "part time"
Integrator Band (EU)	integrator. After the Axis
Limit (%FS)	In Position function has been TRUE for a time equal to Delay, the remaining position error (within the Integrator Band) is integrated. The integrator output is limited to a maximum of plus or minus the Integrator Limit.

```
AXIS[  1]          ↑ QUIT    F1
                             F2
Kp                   SAVE    F3
[0.0000    ]       ↓ <EDT    F4
```

```
AXIS[  1]          ↑ QUIT    F1
                             F2
Ki                   SAVE    F3
[0.0000    ]       ↓ <EDT    F4
```

```
AXIS[  1]          ↑ QUIT    F1
                             F2
KD                   SAVE    F3
[0.0000]           ↓ <EDT    F4
```

```
AXIS[  1]          ↑ QUIT    F1
INTEGRATOR           AXIS    F2
DELAY  (SEC)         SAVE    F3
[0.0999    ]       ↓ <EDT    F4
```

```
AXIS[  1]          ↑ QUIT    F1
INTEGRATOR           AXIS    F2
BAND  (EU)           SAVE    F3
[10.0000    ]      ↓ <EDT    F4
```

```
AXIS[  1]          ↑ QUIT    F1
INTEGRATOR           AXIS    F2
LIMIT  (%FS)         SAVE    F3
[3.1251    ]       ↓ <EDT    F4
```

Figure 3-7. Configuration Axis Gains Screens

CFG–System Parameters

In the CFG mode, use Up/Down arrows to get Configuration System Parameters screen. Push **Edit-F1**. The screen reached depends on the last screen used. Arrow keys move between screens.

```
SYSTEM             ↑ QUIT    F1 QUIT  to Previous Screen
LEVEL 1                      F2
SECURITY CODE        SAVE    F3 SAVE Code to RAM Memory
[0            ]    ↓ <EDT    F4 EDIT Value (0 to 65535)
```

```
SYSTEM             ↑ QUIT    F1 QUIT  to Previous Screen
 COMM PORT 2                 F2
SET UP                       F3
                   ↓ EDIT    F4
```

```
RS232              ↑ QUIT    F1
                             F2
BAUD RATE            SAVE    F3
[9600      ]       ↓ <EDT    F4
```

```
RS232              ↑ QUIT    F1
                             F2
WORD LENGTH          SAVE    F3
[8            ]    ↓ <EDT    F4
```

```
RS232              ↑ QUIT    F1
                             F2
STOP BITS            SAVE    F3
[1            ]    ↓ <EDT    F4
```

```
RS232              ↑ QUIT    F1
                             F2
PARITY               SAVE    F3
[NONE      ]       ↓ <EDT    F4
```

Figure 3-8. CFG-System Parameters

System Parameters

These affect system operation, both axes.

Security Code 1 (Access Modes Past Monitor)

F1 Quit
F3 Save, Saves to RAM Memory
F4 Edit, Edits the value, Value Entry Screen (0 to 999
0 = no protection

RS232 Setup

F1 Quit, Return to previous screen
F4 Edit, Go to Section 3.2.5.1.

RS232 Setup

There are four screens. All four work alike. Any of the four can be used first. Arrow keys move the screen selection up or down.

```
RS232              ↑ EDIT   F1
                            F2
BAUD RATE            SAVE   F3
[9600     ]        ↓ EDT    F4

RS232              ↑ EDIT   F1
                            F2
WORD LENGTH          SAVE   F3
[8        ]        ↓ EDT    F4

RS232              ↑ EDIT   F1
                            F2
STOP BITS            SAVE   F3
[1        ]        ↓ EDT    F4

RS232              ↑ EDIT   F1
                            F2
PARITY               SAVE   F3
[NONE     ]        ↓ EDT    F4
```

Figure 3-9. CFG – RS232 Set-Up

F1 Quit, Return to previous screen
F2 Save, Saves to RAM Memory
F4 Edit, Edits the value

Use arrow keys to get to another screen.

Baud Rate Default = 9600, Available: 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400.

Data Bits Available values are: 7 or 8.

Stop Bits Available values are: 1 or 2.

Parity Available values are: Odd, Even, None.

A common setting is: 9600, 8, 1, None.

The devices at each end of the RS232C communications link must have the same settings.

Digital Input/Output Set-Up

For normal operation it is not necessary to set-up I/O address assignments. The default address assignments for the front panel input and output are as follows:

Address	Name	Description/Function
0	TQ	Output - drive torque (current) enable
1	VEL	Output - drive speed enable
2	MTN	Output - motion in progress
3	IP	Output - in position
4	HC	Output - home sequence complete
5	ERR	Output - error
6	O	Output - uncommitted
7	O	Output - uncommitted
8	DRV	Input - drive amplifier "OK"
9	JG+	Input - jog direction +
10	JG–	Input - jog direction -
11	LIM+	Input - travel limit +
12	LIM–	Input - travel limit -
13	HOM	Input - home switch
14	RUN	Input - run program
15	I	Input - run home sequence

Table 3-1A. Axis 1 (X) Connection Pins – D32 through D2

Address	Name	Description/Function
16	TQ	Output - drive torque (current) enable
17	VEL	Output - drive speed enable
18	MTN	Output - motion in progress
19	IP	Output - in position
20	HC	Output - home sequence complete
21	ERR	Output - error
22	O	Output - uncommitted
23	O	Output - uncommitted
24	DRV	Input - drive amplifier "OK"
25	JG+	Input - jog direction +
26	JG–	Input - jog direction -
27	LIM+	Input - travel limit +
28	LIM–	Input - travel limit -
29	HOM	Input - home switch
30	RUN	Input - run program
31	I	Input - run home sequence

Table 3-1B. Axis 2 (Y) Connection Pins – D32 through D2

Use of I/O address assignments and function references is described in detail in Chapter 4 and in the Programming Reference Manual.

An introduction to I/O and function reference reassignment is given below. There are 256 internal I/O addresses. The first 32 (0 thru 31) are normally assigned to the front panel default functions.

A few internal I/O's serve a fixed function:

255 Always TRUE (1)
254 Always FALSE (0)
253 Output dummy address (Output functions assigned to this address are ignored)

These bit addresses can be used in any I/O logic equation or for address assignments. The following describes how to do this.

In addition to the normal functions assigned to front panel I/O shown in Table 3–1A and 3–1B the controller has several other functions assigned to internal addresses. These functions can be accessed externally by re–assigning the function to a front panel, input or output.

The FUNCTION list and default assignment for each axis is shown in Tables 3-2A and 3-2B.

AXIS 1

Function		Default Address Assignment
OUTPUTS		
Torque Enable		0
Velocity Enable		1
In Motion		2
In Position		3
Home Complete		4
Error		5
Program Select		253
Program Select Bit #	0	253
”	1	253
”	2	253
”	3	253
”	4	253
”	5	253
”	6	253
”	7	253
”	8	253
”	9	253
”	10	253
”	11	253
”	12	253
”	13	253
”	14	253
INPUTS		
Drive Ready		8
Jog+		9
Jog–		10
Limit+		11
Limit–		12
Home Switch		13
Run Program		14
Home Request		15
Program Select		254
Program Select Bit #	0	254
”	1	254
”	2	254
”	3	254
”	4	254
”	5	254
”	6	254
”	7	254
”	8	254
”	9	254
”	10	254
”	11	254
”	12	254
”	13	254
”	14	254
Set Position Reference		254

Table 3-2A. Axis 1 I/O Address Assignments

AXIS 2

Function		Default Address Assignment
OUTPUTS		
Torque Enable		16
Velocity Enable		17
In Motion		18
In Position		19
Home Complete		20
Error		21
Program Select		253
Program Select Bit #	0	253
”	1	253
”	2	253
”	3	253
”	4	253
”	5	253
”	6	253
”	7	253
”	8	253
”	9	253
”	10	253
”	11	253
”	12	253
”	13	253
”	14	253
INPUTS		
Drive Ready		24
Jog+		25
Jog–		26
Limit+		27
Limit–		28
Home Switch		29
Run Program		30
Home Request		31
Program Select		254
Program Select Bit #	0	254
”	1	254
”	2	254
”	3	254
”	4	254
”	5	254
”	6	254
”	7	254
”	8	254
”	9	254
”	10	254
”	11	254
”	12	254
”	13	254
”	14	254
Set Position Reference		254

Table 3-2B. Axis 2 I/O Address Assignments

253 = Ignore Outport 254 = Always Low 255 = Always High

The purpose of each function is described in the Programming Reference Manual.

Suppose your application does not require use of Limit+ or Limit– and you want to use one of these inputs for a trigger on a particular motion segment with a profile.

Further – Axis 1 Limit+ will be used. It is desired that one of the segments within Axis 1 profile requires that this external signal be ‘On’ before the segment executes. The following program steps are required.

- In the CFG mode use Up/Down arrows to go to I/O assignment screen.

CONFIGURATION	↑ EDIT	F1
AXIS[1]	AXIS	F2
I/O ADDRESS	RSET	F3
ASSIGNMENTS	↓ LIST	F4

(Be sure Axis [1] is displayed)

- Push **EDIT–F1**
- Use Up/Down arrows (↑ ↓) to get to the screen showing Limit+.

AXIS[1] INPUT	↑ QUIT	F1
LIMIT +	AXIS	F2
STATE (0)	SAVE	F3
ADDRESS [11]	↓ <EDT	F4

This displays the current address of Limit+ and its state.

- Push **<EDT–F4**.
- Push ‘CLEAR ENTRY’ key.
- Enter ‘254’ then push ‘ENTER’ key, then push **SAVE–F3**.
- The screen now reads:

AXIS[1] INPUT	↑ QUIT	F1
LIMIT +	AXIS	F2
STATE (0)	SAVE	F3
ADDRESS [254]	↓ <EDT	F4

The Limit+ function for Axis 1 is now always set to 0. Input address 11 is now free to use in any logic (event) equation.

The Programming Reference Manual provides detailed information about I/O assignment and functions.

FLASH Memory

FLASH Memory is the memory segment where user programs are saved. It is non–volatile memory, thus it is saved during power off. The User must tell the controller when to save to memory. When you are entering a program (CFG, PRG, VAR modes) each time you hit a ‘Save’, the data is saved to RAM memory, but **not** to FLASH Memory; therefore, when you are satisfied with the programming, you **must** go to the FLASH Memory screen and ‘Save’. It is a good idea to go to this screen several times during a programming session to save the current data so that it is not inadvertently lost during a power down or loss.

In the CFG mode, use Up/Down arrows to go to Configuration FLASH Memory screen.

CONFIGURATION	↑ SAVE	F1
		F2
		F3
FLASH MEMORY	↓ LOAD	F4

Push ‘Save’ to save contents of current programs.

Push ‘Load’ to load FLASH Memory into RAM.

Quick Set Up

The following parameters should be checked or entered prior to attempting any ‘motion’ commands. The screens are shown for reference. The following section describes how to use them.

- Engineering units (counts per desired length unit for your sensor i.e., for 1000 cts/in – Enter 1000).

AXIS[ ]	↑ QUIT	F1
POSITION	AXIS	F2
ENG UNIT	SAVE	F3
[1000.0000]	↓ <EDT	F4

- Select engineering unit name (i.e. Inch, Feet, Millimeters, Centimeters, Revolution, Engineering Units).
- Set software position limits (Maximum Position, Minimum Position).

AXIS[ ]	↑ QUIT	F1
MAXIMUM	AXIS	F2
POSITION (IN)	SAVE	F3
[12345.0000]	↓ <EDT	F4

- Set Maximum Position Error to a value equal to 10% of full travel. This can be adjusted later after the gains have been optimized.
- Set Maximum Speed to within the specification or your system.

AXIS[ ]	↑ QUIT	F1
MAXIMUM	AXIS	F2
SPEED (IN/SEC)	SAVE	F3
[45.0000]	↓ <EDT	F4

AXIS[ ]	↑ QUIT	F1
MAXIMUM POS	AXIS	F2
ERROR	SAVE	F3
[30.0000]	↓ <EDT	F4

- Enter a value for Jog that is fairly slow for a start (10% of maximum speed).

AXIS[ ]	↑ QUIT	F1
JOG SPEED	AXIS	F2
VALUE (IN/SEC)	SAVE	F3
[5.0000]	↓ <EDT	F4

- Enter Maximum Acceleration and Deceleration values within the specification of your system. (Start with 2 times the value of Maximum Speed and adjust to your requirements).
- Enter a Home Speed that is fairly slow for a start (10% of Maximum Speed).

AXIS[ ]	↑ QUIT	F1
MAXIMUM	AXIS	F2
ACCEL (IN/SEC^2)	SAVE	F3
[300.0000]	↓ <EDT	F4

AXIS[ ]	↑ QUIT	F1
HOME	AXIS	F2
SPEED (IN/SEC)	SAVE	F3
[1.0000]	↓ <EDT	F4

- Enter a ‘Kp’ proportional gain of about 1000 – 5000 to start.

```
AXIS[   ]           ↑ QUIT     F1
                               F2
Kp                    SAVE     F3
[5000.0000    ]     ↓ <EDT     F4
```

Don’t forget to push ‘SAVE’ on each screen and also to “SAVE to FLASH Memory” after you have entered all parameters. (Go to CFG mode and get Configuration FLASH Memory screen.)

Testing Initial Set–Up

After you have entered and checked the configuration parameters, operation of the controller can be tested by using the Jog function. The following procedure is performed:

- Apply power to the axis actuators.

WARNING
If one or both axes begin to move, quickly remove power from actuators to prevent machine damage or personal injury. – Recheck all wiring and configuration parameters. – Refer to Troubleshooting page ##.

- Press RUN mode key and refer to Figure 3-10 for the screens available.

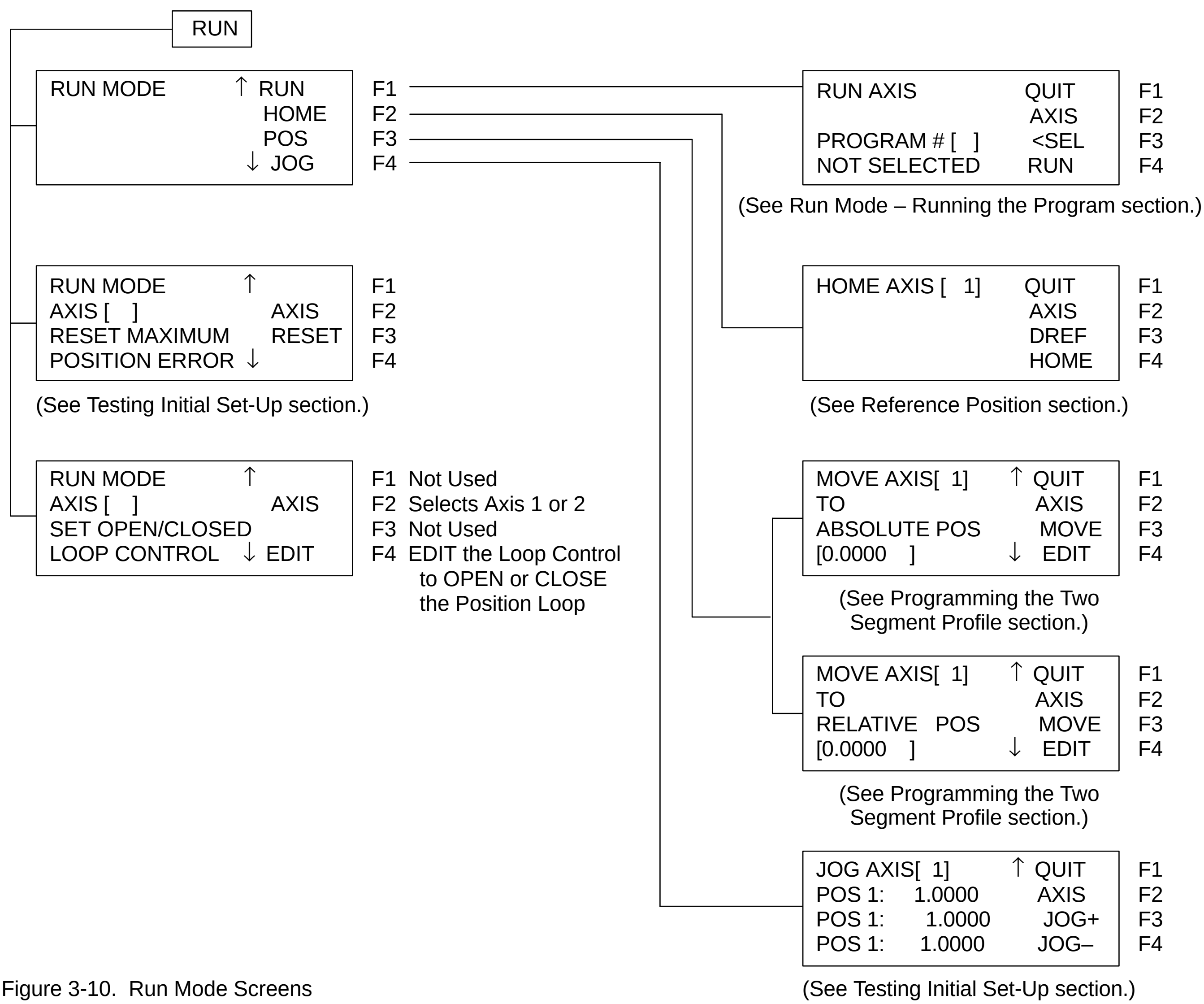

Figure 3-10. Run Mode Screens

From the main screen, press **F4** key to go to Jog function.

Then press **F4** key or **F3** key to jog the axis in the '–' or '+' direction. Press **AXIS–F2** to change axis. The actuator should move. If the motion is sluggish or unstable, it may be because the axis gain factors are not set properly. However, if motion is obtained, proceed with the set-up procedures. If the axis does not move, it is possible that the error is set for various reasons. Push the Up/Down arrow keys to reach the Error Reset screen:

RUN MODE ↑		F1
AXIS [ ]	AXIS	F2
RESET MAXIMUM	RESET	F3
POSITION ERROR ↓		F4

Press **RSET–F3** key then go back to Jog screen by pushing the Up/Down arrow keys. If the axis still does not move, refer to Troubleshooting page ##.

Reference Position

There are several methods available to provide a positional reference for the axes.

Automatic Home Sequence

The automatic homing sequence can be used as follows:

The automatic homing sequence requires at least one external input for each axis – Input 4/20 – **HOM** for the home location limit switch. It also requires that the encoder have a "C" channel 'marker' input. The homing sequence can be started by activating Home Request input 15/31 from the terminal as follows:

Figure 3-11. Home Switch Connections

In the CFG mode press **F2** to access Home Axis screens. There are four configuration parameters associated with the home sequence:

Home Position – defines the **Home Position** value. (Can be any number within the controller position range)

AXIS[ ]	↑ QUIT	F1
HOME	AXIS	F2
POSITION (EU)	SAVE	F3
[0.0000]	↓ <EDT	F4

Home Offset – defines the **offset** position from **Home Limit Switch** in the length units previously selected (inches, meters, etc).

AXIS[ ]	↑ QUIT	F1
HOME	AXIS	F2
OFFSET (EU)	SAVE	F3
[0.0000]	↓ <EDT	F4

During the homing sequence, the axis will move to the home limit switch, then move the **offset** amount, where it will set position value equal to **Home Position**.

Home Speed – Defines the speed that the axis will move to the home limit switch.

AXIS[ ]	↑ QUIT	F1
HOME	AXIS	F2
SPEED (EU/SEC)	SAVE	F3
[199999 9694]	↓ <EDT	F4

Home Direction – Defines the direction the axis will move to find the home limit switch. [Advance or Return]

AXIS[ ]	↑ QUIT	F1
HOME	AXIS	F2
DIRECTION		F3
[RETURN]	↓ <SEL	F4

Note: Be sure to save to FLASH Memory after you have entered the home values; then the machine can always be 'homed' to the same position after a power down.

When the homing sequence is activated, the axis moves at Home Velocity (Configure mode acceleration and deceleration is active) until the home limit switch is sensed, then the axis decelerates to zero, reverses direction and moves at Creep Velocity until the first encoder marker or resolver zero is detected **after** moving off the home limit switch. The axis decelerates to zero and moves back to the marker position. At this time, the 'Home Complete' output (HC) is turned 'On' and stays 'On' until power is 'Off' or a reset occurs. This is the reference position and it is set to the value defined in the Home Position command in the CFG mode. (0.0000 or any position in range). All absolute positions are now referenced to this position. If home offset is set to other than zero, the axis will move to the offset value.

Manual Home Function

Another method for changing the position reference is the ‘DREF’ command (Define Reference). The axis should first be jogged to the desired home position. This command is available in the RUN mode on the Home screen:

HOME AXIS [1]		
	QUIT	F1
	AXIS	F2
	DREF	F3
	HOME	F4

Press the **F3** key. The position reference will now be set to the home position value as set in the CFG mode.

For incremental feedback transducers such as encoders, the ‘DREF’ position reference is not saved. When the unit is shut down and then powered back ‘On’, you must jog to a desired position and then do ‘DREF’ to re-reference the position.

The ‘DREF’ function can be done automatically in programs or any time by use of a digital input. The internal function ‘Set Position Reference’ can be assigned to one of the front panel input addresses (page ##). Refer to Programming Reference Manual to learn how to use this function in programs. Use of the ‘Set Position Reference’ function can be used as a Home; however, the axis must first be physically located at the desired home position.

Axis Limits

Axis plus (+) and minus (–) travel limits should be installed because damage or improper operation can occur from fully extended or retracted actuator operation.

CAUTION

Use of Limit+ and Limit– travel limits, depend on a correctly operating SMC20 controller.

If machine damage or personal injury can result from overtravel, it is recommended that back up limit switches located outside the LIMIT+, LIMIT– switches be installed and wired such that actuator power (electric or hydraulic) IS REMOVED when the limit switch is tripped. The machine must then be physically reset to an acceptable position.

When either limit switch is turned ‘On’, the axis will decelerate (at current deceleration value) to zero velocity and hold position. The axis must then be jogged off the limit switch in the opposite direction before normal operation can resume.

Set-Up Checklist and Operation Test

It is recommended that the following checklist and operation test be performed before entering or running programs. This test verifies that the system wiring is correct and that the controller operates correctly.

1. AS A MINIMUM, the following axis parameters should be set to reasonable values for your system.
   - – Engineering units (feedback sensor counts per engineering units)
   - – Maximum move velocity
   - – Maximum Acceleration/ Deceleration
   - – Home Parameters (only if home sequence is used]
   - – Maximum position error
2. Axis gains are set as a minimum, a Kp Gain should be set to 1000 – 5000 to obtain motion.
3. Emergency stop provisions connected and tested.
4. Limit+ and Limit– connected if being used.
5. Jog motion has been operated and verified.

Table 3-3. Set-Up Checklist

Operational Test – The following test will verify that the controller is wired correctly and is operating. Basic positioning will be accomplished.

Performance of this test assumes that:

Jog motion has been operated and verified.

- Installation checklist is completed.
- Set-Up checklist is completed.
- Jog motion for each axis has been tested.
- All parameters have been SAVED.
- Operator has read this manual.

1. Power ‘On’ – go to RUN mode.

   Go to Jog screen and move axis to approximate center position (inside limits or minimum and maximum travel). Record the axis position __________.

2. Push **Quit–F1** key and then **POS–F3** key to reach the following screen:

MOVE ANS[1]	↑ QUIT	F1
TO	AXIS	F2
ABSOLUTE POS	MOVE	F3
[0.0000]	↓ <EDT	F4

   Push **<EDT-F4** key and enter in a position value to move to – which is inside the travel range. The value should be different than the one recorded above. When the **Save-F3** key is pressed, the above screen will return (with the position value that was entered).

3. Press **Move–F3** key. The axis should move to the position entered above.

   If the axis motion is unstable or exact position is not reached or motion is sluggish, it may be because the proportional or other gains are not optimum. Refer to Troubleshooting section (page ##) for tuning and troubleshooting help.

4. If homing sequence is implemented, perform the following check:

   Go to Run mode main screen and select **‘Home’**. Press **Home–F4** key. Axis should move to the home limit switch, stop, move to offset position and just off the switch and stop. The axis position should now be the programmed Home Position (usually 0.000)

5. If Limit+ and/or Limit– functions are used, perform the following check:

   Go to Jog screen and move axis towards one of the limits. When a limit switch is reached, the axis should decelerate to zero velocity (stop). Continued pressing of the same Jog key should not cause any motion. Press the other Jog key. The axis should move towards the opposite limit switches. Keep jogging in this direction until the limit switch is reached. Again, the axis should stop.

Table 3-4. Operational Test

Programming

Introduction

Efficient programming requires that several steps be performed in sequence. These are:

1. **Determine overall motion limits on each axis** — the axis configuration limits. These will be entered for each axis from the Configuration (CFG) mode. The controller will use these limits as defaults at other programming levels.

   These configuration parameters should be carefully verified. Jog each axis between the extreme axis position limits, or the maximum positions that will ever be programmed. Test Maximum Velocity, Acceleration, and Deceleration rates, using Position screen. (Ref: Operation Test section, page ##.)

WARNING

Motion profile programming should proceed only after these tests are completed and test results or values lower than test results are entered in the CFG mode for each axis. See Initial Set-Up section (page ##) for an explanation of the Terminal screens used for entry.

2. **Determine what axis profile(s) are to be programmed.** Choose a program number for each. Program numbers can range from 1 to 999 and don’t have to be sequential.

3. **Determine what the limit values for the motion parameters** should be for each profile. If these are the same as the configuration axis limits, they do not have to be entered. However, it is likely that if several profiles are programmed, some should have reduced limit values. These individual program limit values default to the axis configuration values. In turn, the motion parameters in each segment have the program values as defaults.

4. **Enter the program limits for the first profile to be programmed.** These are entered from the Program mode (PRG) – described in this chapter.

5. Now start programming the segments of the first profile. When a program is finished (and more often if the task is long) save to FLASH Memory.
6. Repeat from step 4 to program additional profiles. Be sure to be on the proper axis and be sure to select a new program number each time.

   For references:
   [Axis 1 is also called Axis X]
   [Axis 2 is also called Axis Y]

Profile Definition

A motion profile is a description of the motion of an axis over a complete cycle of motion. Typically the description consists of a graph of velocity (Y coordinate) vs. position (X coordinate) and a list of notes related to specific positions or events which might occur during motion.

Figure 4-1. Typical Motion Profile

The above profile is made up of three segments; a move to position 20 in. at 30 in./sec. velocity, decelerate to 2 in/sec. The second segment continues at 2 in/sec to 30 in. The third segment returns to 0 in. at 30 in/sec.

A profile could be as simple as an incremental move (one segment).

Segment Definition

One segment can define the following: Entry Conditions, Motion Parameters, Exit Conditions, Jump Conditions, Gains and Variables.

1. **Entry Conditions.** This has 2 parts:
   a) Dwell Time. This is a time in seconds that the controller should wait after completion of the preceding segment. A zero value usually applies to all but the last segment of motion in one direction.
   b) Input Event Conditions. One or more logical equations to be evaluated at the end of Dwell Time. This logic can inhibit the axis until specified conditions are met.
2. **Motion Parameters**
   a) Absolute or Relative Target Position
   This is the position to be reached at the end of a segment.
   b) Acceleration/Deceleration
   These are the values to be used for velocity changes in the segment.
   c) Move Velocity
   This velocity is intended to be used for the requested change in position from Entry Position to Target Position. After Dwell Time (if any) and Input Event = TRUE (if any), the axis will attempt to go from the input velocity (Target Velocity of previous segment) to Move Velocity. If Move Velocity can be reached before it is necessary to decelerate, motion is carried out at that velocity until deceleration starts.
   d) Target Velocity
   This is the velocity at which the axis is to be moving at the end position of the segment. The position at which deceleration should start is computed by the controller. A short move with a high Move Speed may therefore never actually reach Move Speed because of the start of deceleration (or acceleration) to reach Target Velocity. Note that Target Velocity can be 0 or any velocity less than the Profile Maximum Velocity. Having a Target Velocity as part of the segment allows the controller to compute when to start acceleration or deceleration from the Move Velocity to reach the desired Target Point Velocity rather than having the programmer do this computing, and it tends to keep the transition from one segment to another free of acceleration spikes.
3. **Exit Condition**

   Conditional exiting is accomplished with the Exit Event equation. This is a Logic Output Event description, which can be a series of equations evaluated from top to bottom, as entered from the Hand-Held Terminal. The Output Event is evaluated when the Position Reference output has reached the Target Position. That is: Position Reference = Target Position. At this time the slide may not have reached the Target Position because of following error. If the next segment calls for motion in the reverse direction and neither the Exit Event or the Jump Event meet the test for In Position, the axis may reverse motion without going all the way to Target Position. (The In Position signal becomes TRUE when the feedback position matches the Target Position within a tolerance—the axis tolerance is set in the CFG mode and a lower program value can be set in the PRG mode. The In Position is evaluated in every segment and thus can be triggered ‘on the fly’.

   Note that some judgement must be exercised in setting the In Position tolerance. If the machine capability is barely good enough to satisfy the programmed tolerance, the program may occasionally stop or slow considerably while waiting for the tolerance to be satisfied. When it is allowable for an axis to run past a theoretical endpoint, it may be better to program the Target Position a little past the theoretical endpoint position and open the In Position tolerance.

   The Exit Event may test multiple conditions and may use several equations. Some or all of these equations may output a logic TRUE or FALSE signal to an internal flag (I/O location). The flag may then be tested by another segment or by the other axis.
4. **Jump Conditions**

   There are two conditions:

   a) Unconditional Jump. This would most likely be used to continuously run a profile or group of segments as long as 'Run' is commanded. A jump is unconditional if a destination is programmed, but no logic equation is entered. For example,

```
SEGMENT # (2  )      ↑ QUIT    F1
                       <DEL    F2
JUMP TO                 SAVE   F3
SEGMENT # [1  ]      ↓ <SEL    F4
```

b) Conditional Jump. Defined by a logic equation evaluated after the Exit Conditions are satisfied. For example, from the Jump Event Condition screen: —> EDIT and ENTER.

```
                              ↑ QUIT   F1
 1   START                       DEL   F2
 2    [10#[13]      ]           <INS   F3
      END EQ                  ↓ SAVE   F4
```

Note that not all of the above four categories have to have entries in every segment. It is possible to program a segment that does not move at all, but only checks logic equation(s) and it is possible to program a simple move with no special Entry or Exit Conditions. A simple move may use several of the axis default values and thus require not much more than an Absolute or Relative Target Position.

There are two additional things that can be, but do not have to be used in a segment:

5. **Gains**

   A maximum of three gains can be defined. These override selected PID gains to change axis stability FOR THIS SEGMENT ONLY. A possible use would be with an axis that has a large change in inertia for a few segments of the profile.

6. **Variables**

   Refer to Programming Reference Manual.

Sequence of Execution

When the SMC20 executes a segment, it does not do all of the operations at the same time. There is an order, or sequence, in which execution occurs. It is important to keep this order in mind as results different than otherwise expected can occur.

1. If a Dwell Time has been programmed as part of the Input Conditions, it will start timing and completely time out before anything else occurs.

2. Next, any Input Events that are part of the Input Conditions are executed – such as reading inputs or turning ‘On’ or ‘Off’ any outputs.

3. Next, Motion Parameters and Motion Profile Data are executed – all at the same time. This includes reading any variables required, setting new values of gains for that segment and computing and loading all targets.

4. Next, Output Events that are part of the Output Conditions – such as turning Outputs ‘On’ or ‘Off’ and setting internal flags — are executed.

5. Next, any Conditional Jumps that are part of the Output Conditions are executed.

6. And finally, any Jump Conditions are executed — direct jumps are those with no conditions attached, such as jumping directly to a particular segment number no matter what else is happening.

Note, that if gains or variables are updated at any time prior to step 3, they will take effect in the **current** segment – otherwise they will take effect in the following segment.

Program Mode

The PRG mode is used to enter new segments, edit previously entered segments AND to enter changes to parameters that will apply ONLY to segments. The parameters changed are those entered in the CFG mode (See Introduction to Programming section, page ##). These parameters can only be changed in a downward direction in the PRG mode. The PRG mode is entered by pressing the **‘PRG’** key on the pendent.

```
PRG
```

```
PROGRAM MODE      ↑ NEW    F1
AXIS[ 1]            AXIS   F2
PROGRAM#[    ]      <SEL   F3
DOES NOT EXIST    ↓ EDIT   F4
```

```
PROGRAM MODE      ↑ COPY   F1
AXIS[ 1]            SEG    F2
PROGRAM#[    ]      <DEL   F3
DOES NOT EXIST    ↓ LIST   F4
```

```
PROGRAM#(   )          QUIT   F1
                              F2
PROG PARAMETERS        EDIT   F3
PROGRAM SEGMENTS EDIT         F4
```

```
PROGRAM MODE      ↑ NEW    F1
AXIS[ 1]            AXIS   F2
PROGRAM#[    ]      <SEL   F3
DOES NOT EXIST    ↓ EDIT   F4
```

```
PROGRAM MODE      ↑ COPY   F1
AXIS[ 1]            SEG    F2
PROGRAM#[    ]      <DEL   F3
DOES NOT EXIST    ↓ LIST   F4
```

```
PROGRAM#(   )        ↑ QUIT   F1
SEGMENT#(   )                 F2
 VARIABLES                    F3
                     ↓ EDIT   F4
```

```
PROGRAM#(   )        ↑ QUIT   F1
SEGMENT#(   )                 F2
 ENTRY                        F3
CONDITIONS           ↓ EDIT   F4
```

```
PROGRAM#(   )        ↑ QUIT   F1
SEGMENT#(   )                 F2
 MOTION                       F3
PARAMETERS           ↓ EDIT   F4
```

```
PROGRAM#(   )        ↑ QUIT   F1
SEGMENT#(   )                 F2
 EXIT                         F3
CONDITIONS           ↓ EDIT   F4
```

```
PROGRAM#(   )        ↑ QUIT   F1
SEGMENT#(   )                 F2
 JUMP                         F3
CONDITIONS           ↓ EDIT   F4
```

```
PROGRAM#(    )      QUIT   F1
                           F2
AXIS                SAVE   F3
[ 1   ]             <SEL   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MAXIMUM                   F2
POSITION (EU)       SAVE   F3
[2147483647.00 ]  ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MINIMUM                   F2
POSITION (EU)       SAVE   F3
[-2147483648.0 ]  ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MAXIMUM                   F2
SPEED (EU/SEC)      SAVE   F3
[399999.9388 ]    ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MAXIMUM                   F2
ACCEL (EU/S^2)      SAVE   F3
[199743.1951 ]    ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MAXIMUM                   F2
DECEL (EU/S^2)      SAVE   F3
[199743.1951 ]    ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 MAXIMUM                   F2
ERROR (EU2)         SAVE   F3
[2000.0000 ]      ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
 IN POS                    F2
ERROR (EU2)         SAVE   F3
[10.0000 ]        ↓ <EDT   F4
```

```
PROGRAM#(    )    ↑ QUIT   F1
                           F2
EVENT               <DEL   F3
CONDITION         ↓ EDIT   F4
```

Figure 4-2. Program Mode

Use arrow keys to shift between screens.

Upper left screen. Figure 4–2.

A) For existing programs:

F1 New, Not Used

F2 Axis, Select Axis 1 or 2

F3 <SEL, Select program

F4 Edit, Edit (second level) The “Does Not Exist” message will go away when an existing program is selected.

B) For a new program:

F1 New, The Program Entry screen appears. Enter a program number. Push ENTER to end entry screen. If a program with the same number already exists, an error screen appears. Push ENTER to leave the error screen. Try again starting with **F1–New**. When new number has been chosen, proceed as for an old program.

a) The program number must have been selected from the upper screen.

Note: The program number must have been selected from the upper screen to proceed.

F1 Not Used

F2 Axis, Selects the axis (but not the program)

F3 <DEL, Deletes the program shown on the axis shown

F4 List, Lists the program to the RS232C port. It is suggested that any program to be deleted first be listed, since all or part of that program might be a useful reference.

Program

Push **Edit–F4** on the top left screen to get the next level screen.

F1 Quit, Return to previous screen

F2 Not Used

F3 Edit, Program Parameters
Program Parameters are to the individual program what Axis Parameters (CFG mode) are to all the programs on an axis. These should be programmed before the segments, as these values act as defaults for the segments.

F4 Edit, Program Segments
A program consists of one or more segments. Remember, a program consists of the instructions needed to carry out one motion profile. A motion profile is divided into segments for programming purposes. Program segments may not necessarily correspond 1 to 1 with profile segments as special purpose program segments can be used to carry out non-motion aspects of the profile such as waiting for specific input logic to be satisfied.

PRG – Program Parameters

See Figure 4-3. Program Parameters are to a program what Configuration Parameters are to an axis. These parameters all have the Configuration Parameter values as defaults so it may only be necessary to enter a few values. These program parameter values in turn become defaults and limits for segment parameter values.

Note: As with the Configuration Parameters, changes after segments are programmed WILL NOT result in automatic changes in the segments. When the program runs out of limit segment values, based on the newest configuration and program parameters, it will result in error messages.

Screens in Figure 4-3:

Axis

F1 Quit, Return to previous screen

F2 Not Used

F3 Save, Saves axis selection to RAM Memory

F4 <SEL, Select Axis [1 (X) or 2 (Y)]
Selection on this screen reassigns the entire program to the axis selected and removes it from the previously selected axis. This is NOT Copy, but reassignment.

```
PROGRAM#(    )      QUIT    F1
                            F2
AXIS                SAVE    F3
[ 1   ]             <SEL    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MAXIMUM                    F2
POSITION (EU)       SAVE    F3
[2147483647.00 ]  ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MINIMUM                    F2
POSITION (EU)       SAVE    F3
[-2147483648.0 ]  ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MAXIMUM                    F2
SPEED (EU/SEC)      SAVE    F3
[399999.9388  ]   ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MAXIMUM                    F2
ACCEL (EU/S^2)      SAVE    F3
[199743.1951  ]   ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MAXIMUM                    F2
DECEL (EU/S^2)      SAVE    F3
[199743.1951  ]   ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 MAXIMUM                    F2
ERROR (EU2)         SAVE    F3
[2000.0000  ]     ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
 IN POS                     F2
ERROR (EU2)         SAVE    F3
[10.0000  ]       ↓ <EDT    F4
```

```
PROGRAM#(    )    ↑ QUIT    F1
                            F2
EVENT               <DEL    F3
CONDITION         ↓ EDIT    F4
```

Figure 4-3. Program Parameters

The next several screens all operate the same.

MAXIMUM POSITION (EU)
MINIMUM POSITION (EU)
MAXIMUM SPEED (EU/SEC)
MAXIMUM ACCEL (EU/S^2)
MAXIMUM DECEL (EU/S^2)
MAXIMUM POSITION ERROR (EU)
IN POSITION ERROR (EU)

The Engineering Unit selected for the axis (CFG mode) is substituted for EU on the actual screen.

F1 Quit, Return to previous screen

F2 Not Used

F3 Save, Push after leaving Value Entry screen to save to RAM Memory

F4 <EDT, Edit Value Entry screen

Event Condition

F1 Quit, Return to previous screen

F2 Not Used

F3 <DEL, Delete a previously programmed event

F4 <EDT, Edit, This brings up a separate lower level screen. See the section on Logic Equations (page ##).

PRG – Segments

Use arrow keys to select the desired screen.

Upper Screen

F1 Quit, Moves back to previous

F2 <SEL, Works if there are segments to select.

F3 NEW, If the segment box is empty (new program) or if segments are to be added at the end of a program, pushing New brings up a new block number, one bigger than the previous highest number.

F4 <EDT, Edit, Goes to the Segment Edit screen (Fig. 4–5)

Lower Screen

F1 Quit, Moves back to previous

F2 <Del, Deletes a segment and increments down the number of any segments which previously had higher segment numbers.

F3 INS, Inserts a segment ahead of the segment shown. The new segment uses the number on the screen and the segment which previously had that number and all higher number segments have their number incremented up.

F4 List, List segment to RS232C port.

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]       <SEL   F2
                     NEW   F3
DOES NOT EXIT     ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]       <DEL   F2
                     INS   F3
DOES NOT EXIT     ↓ EDIT   F4
```

Figure 4-4. PRG Segments

Segment Parameters

There are six groups of Segment Parameter entries corresponding to the six screens shown in Figure 4-5. These are reached when **F4** is pushed at the segment selection (Figure 4-4).

Programming some segments requires use of all six screens at this level. All should be examined when any segment is programmed or edited. The screen which comes up first depends on which was last accessed. Use arrow keys to change screens. Use the screens in any order. Use of these screens will be explained in the following examples.

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]              F2
 VARIABLES                 F3
                  ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]              F2
 ENTRY                     F3
CONDITIONS        ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]              F2
 MOTION                    F3
PARAMETERS        ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]              F2
 GAINS                     F3
                  ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]       <DEL   F2
 EXIT                      F3
CONDITIONS        ↓ EDIT   F4
```

```
PROGRAM#[   ]     ↑ QUIT   F1
SEGMENT#[   ]              F2
 JUMP                      F3
CONDITIONS        ↓ EDIT   F4
```

Figure 4-5. Segment Parameters

Profile and Segment Programming Example

The following example shows how to program a simple motion. It could represent a simple axis motion – move to a position, dwell for 1 second and return to a predetermined home position.

Figure 4-6. Single Axis Motion

The motion is easily implemented with a two segment profile. Segment 1 starts at rest, at zero position, and accelerates at 100 in/sec^2 (.26g) to a Move Speed of 30 in/sec (1800 rpm). It then decelerates to reach a final position at 30 in.

Segment 2 starts with a wait time of 1 sec, then moves at 40 in/sec to return to zero. Acceleration and deceleration for each axis is set at 100 in/sec^2. Target Velocity in each segment is zero.

Programming the Two Segment Profile

Enter the Configuration Limits (Reference Initial Set-Up section, page ##)

Suppose the actual mechanical motion limits are minus 5 inches and plus 40 inches. It is generally best not to program to the very last possible count, as a mechanical motion that is blocked short of programmed position can develop very high forces. Let us choose to set the Configuration mode Maximum Position at 35.0 inches and the Minimum Position at –4.5 inches.

Mechanical stops:	–5.0, +40.0
Configuration:	Maximum Position 35.0
Motion parameters:	Min. Position –4.5 Max. Velocity 50 Max. Acceleration 300 in/sec^2 Max. Deceleration 300 in/sec^2 Home Position 0
Initial Values to be reduced	Max. Position Error 30. in. Max. In Position Error 1.0 in. Axis Event Condition Undefined Jog Speed 1 in/sec

Go to PRG mode. Assume that no programs have been saved. The following screen should come up:

```
PROGRAM MODE      ↑ NEW   F1
AXIS[   1]          AXIS  F2
PROGRAM#[   ]       <SEL  F3
DOES NOT EXIST    ↓ EDIT  F4
```

Press **NEW–F1**.

```
PROGRAM MODE      ←       F1
AXIS[   1]        →       F2
                          F3
DOES NOT EXIST            F4
```

Enter program number (i.e. 10) and press ‘ENTER’. The following screen will appear:

```
PROGRAM MODE      ↑ NEW   F1
AXIS[   1]          AXIS  F2
PROGRAM#[ 10  ]     <SEL  F3
                  ↓ EDIT  F4
```

Press **EDIT–F4**. The following screen will appear:

```
PROGRAM#[ 10 ]        QUIT  F1
                            F2
PROG PARAMETERS  EDIT       F3
PROG SEGMENTS    EDIT       F4
```

For this example, the program parameters can be the same as the axis parameters, except that we will change the acceleration and deceleration to 100 in/sec^2.

Press ‘Program Parameters’ **Edit–F3**. The following screen will appear:

```
PROGRAM#[ 10 ]     ↑ QUIT  F1
                           F2
AXIS                 SAVE  F3
[ 1]               ↓ SEL   F4
```

Press Up/Down arrows until the following screen appears:

```
PROGRAM#[ 10 ]     ↑ QUIT  F1
 MAXIMUM                   F2
ACCEL (IN/S^2)       SAVE  F3
[300.0000   ]      ↓ <EDT  F4
```

Press **<EDT–F4**, ‘CLEAR ENTRY’, Enter ‘100’, press ‘ENTER’ then **Save–F3**. The following screen will appear:

```
PROGRAM#[ 10 ]     ↑ QUIT  F1
 MAXIMUM                   F2
ACCEL (IN/S^2)       SAVE  F3
[99.6799   ]       ↓ <EDT  F4
```

Important Reminder: The value used by the controller depends on resolution capability of the loop processor. The controller responds to an entry with the nearest exact value that it will run with; e.g., with engineering units set at 1000 counts, entering an acceleration of 100.00 in/sec^2 results in an actual value of 99.6799 in/sec^2.

Repeat above screen for deceleration of 100 in/sec^2.

Press **Quit–F4** to get back to the program screen. The following screen will appear:

```
PROGRAM#[ 10 ]       ↑ QUIT  F1
                             F2
PROG PARAMETERS      EDIT    F3
PROG SEGMENTS       ↓EDIT    F4
```

Press 'Program Segments' **Edit–F4**. The following screen will appear:

```
PROGRAM#[ 10 ]      ↑QUIT | F1
SEGMENT#[    ]       <SEL | F2
                      NEW | F3
DOES NOT EXIST      ↓EDIT | F4
```

Press **NEW–F3**. The following screen will appear:

```
PROGRAM#[ 10 ]      ↑QUIT | F1
SEGMENT#[1   ]       <SEL | F2
                      NEW | F3
                    ↓EDIT | F4
```

Note that Segment is automatically set to 1.

Press **Edit–F4**. The following screen will appear:

```
PROGRAM#[ 10 ]      ↑QUIT | F1
SEGMENT#[1   ]       <SEL | F2
 VARIABLES                | F3
                    ↓EDIT | F4
```

Note: If previous programming was done, the Entry screen may be a different parameter. It will always return to the last screen 'edited'.

Press arrow keys to reach the following screens. Ignore parameters not required for this program. The following screen will appear:

```
PROGRAM#[ 10 ]      ↑QUIT | F1
SEGMENT#[1   ]       <SEL | F2
 MOTION                   | F3
PARAMETERS          ↓EDIT | F4
```

Press **Edit–F4**. The following screen will appear:

```
SEGMENT#[1   ]      ↑QUIT | F1
[ABSOLUTE TARG]      <SEL | F2
POSITION (IN)        SAVE | F3
[0.0000    ]         ↓EDT | F4
```

Press **F2** to select Relative or Absolute position.

Press **<EDT–F4**, then 'CLEAR ENTRY'. Enter '30' and press 'ENTER', then press **Save–F3**. The following screen will appear:

```
SEGMENT#[1   ]      ↑QUIT | F1
[ABSOLUTE TARG]      <SEL | F2
POSITION (IN)        SAVE | F3
[30.0000    ]        ↓EDT | F4
```

Repeat above for each of the following parameters:

Target Speed = 0.0000 in/sec

Move Speed = 30.0000 in/sec

Verify that acceleration, deceleration, are at 100.0000 in/sec^2.

This completes the programming required for Segment 1 for the example. It is a good idea at this point to go to CFG mode and save to FLASH Memory.

Press **'PRG'** key to continue programming of Segment 2. Press Edit–F4, then press 'Program Segments' Edit–F4.

```
PROGRAM#[ 10 ]      ↑QUIT | F1
SEGMENT#[1   ]       <SEL | F2
                      NEW | F3
                    ↓EDIT | F4
```

Press **NEW–F3** to get to Segment 2.

Press **Edit–F4** and use arrow keys to go to the Entry Conditions screen. Then press **Edit–F4** again. Use arrow keys again to go to Dwell Time screen:

```
SEGMENT#[2   ]      ↑QUIT | F1
DWELL                <SEL | F2
TIME                 SAVE | F3
[0.0000    ]         ↓EDT | F4
```

Press **<EDT–F4** and then 'CLEAR ENTRY'. Enter '1.0', then press 'ENTER' and **Save–F3** to set to 1.0000 sec.

Press **Quit–F1**, then use the same procedure as above for Segment 1 to program or verify the motion parameters for Segment 2:

Move Speed:	40.0000 in/sec
Target Speed:	0.0000 in/sec
Absolute Position:	0.0000 in
Acceleration/ Deceleration:	100.0000 in/sec^2

Return to the CFG mode and save the program to FLASH Memory.

I/O Event Programming – Introduction

Digital input and output, internal flags, variables and axis positions all can be used in logical relationships called Events. The logic processing and format is similar to 'Boolean Logic Arithmetic'. Events can be programmed to occur as follows:

- In the CFG mode. Events programmed here are called 'Global Events' because the processing of these events occurs **always** – whenever power is 'On'. Thus, these logic operations work independent of motion profiles, segments, etc.
- In the 'Profile Program Parameters'. These events are called 'Global Events' also because they are active anytime the profile is in the Run mode regardless of whether motion is occurring or not.
- In the 'Segments'. These events are evaluated **only** when the segment is active. The segment has three possible event conditions:
  - – Entry Events
  - – Exit Events
  - – Jump (to another segment) Conditional Events

Programming Example

The previous example is now expanded to include some I/O events.

Example 1: It is desired that an output turn 'On' when the extend position is reached and stay 'On' until the retracted position is reached – use output 6 (06). The address for Output 6 (06) is zero.

Figure 4-7. I/O Example

In PRG mode, go to 'Program Segments' and select Segment 1 (Program #10) Exit Conditions:

```
PROGRAM#[ 10 ]     ↑ QUIT   F1
SEGMENT#[1   ]      <SEL    F2
EXIT                        F3
CONDITIONS         ↓EDIT    F4
```

Press **Edit–F4**.

```
SEGMENT#[1   ]     ↑ QUIT   F1
                     DEL    F2
OUTPUT EVENT                F3
CONDITION          ↓EDIT    F4
```

Press **Edit–F4** again. The following screen should now be up:

```
                   ↑ QUIT   F1
1  START             DEL    F2
2  [END EQ  ]        INS    F3
3  END EQ          ↓ SAVE   F4
```

We will now enter a single logic equation to turn 'On' output 06.

Press **<INS–F3**.

```
SELECT SYM OR      ↑ QUIT   F1
VAL TO INSERT        ACPT   F2
                     SEL    F3
OUTPUT TO          ↓ EDT    F4
```

The Up/Down arrow keys are used to select the options which appear in the bottom row. Press Up/Down arrows until 'OUTPUT TO' is reached, then press **ACPT–F2**. Press **<INS–F3** to get above screen again; select IO#[ ] and press **<EDT–F2**.

```
SELECT SYM OR      ↑   ←    F1
VALUE TO INSERT        →    F2
                            F3
I O#[     ]        ↓        F4
```

Enter '6' and press 'ENTER' key, then press **ACPT–F2**. The display will go back to the first screen above and show as:

```
                   ↑ QUIT   F1
3  I O#[  6]         DEL    F2
4  [END EQ  ]        INS    F3
5  END EQ          ↓ SAVE   F4
```

Note: Address 06 will actually cause the physical output number 1 to turn 'On' (See I/O Address Assignments in Tables 3–1A and 3–1B).

You can use the Up/Down arrow keys to scroll through the equations entered. It should appear as:

1 START
2 OUTPUT TO
3 IO# [06]
4 END EQ

Note: Brackets appear in F3 field to highlight the port to be selected.

Press **Save–F4** to save to RAM. (Be sure to go to CFG mode and save to FLASH Memmory when done).

Now when the axis is extended at the end of Segment 1, the output '06' will go 'On'.

The next step is to program Segment 2, (the retract motion) with an Exit Condition as above, except that the logic equation to enter is:

1 START
2 OUTPUT TO
3 NOT
4 IO# [06]
5 END EQ

Use the same sequence as above to enter the equation. Now when the axis is retracted to the 0.0000 inch position, the output will turn 'Off'.

A logic equation can have up to 256 elements for each event. Also, the equation can include axis positions and variables as arguments. Refer to the Programming Reference Manual to learn how to use all of event logic capabilities.

Jump Instruction

The Jump command is used to program 'branches' or 'repeat' cycles. There are two types of Jumps – an Absolute Jump or a Conditional Jump. Jumps are allowed only in segments and only from one segment to another. Therefore, all branches or loops are within a profile.

The previous example can be programmed to run continuously and repeat the two segment motions as long as the run switch is 'On'. The Jump is programmed as follows:

Go to PRG mode and select 'Program Segments' for Axis 1 Program 10 (as previously programmed) for **'Edit'**. Select (**<SEL**) segment #2, **Edit**, then use arrow keys to select the screen for Jump Conditions.

```
PROGRAM#[ 10 ]     ↑ QUIT   F1
SEGMENT#[2   ]      <SEL    F2
JUMP                        F3
CONDITIONS         ↓EDIT    F4
```

Use the arrow keys to obtain the following screen:

```
SEGMENT#[2   ]     ↑ QUIT   F1
                    <DEL    F2
JUMP TO              SAVE   F3
SEGMENT#[    ]     ↓ <SEL   F4
```

Push **<SEL–F4** to select Segment #1.

SEGMENT#[2]	↑ QUIT	F1
	<DEL	F2
JUMP TO	SAVE	F3
SEGMENT#[1]	↓ <SEL	F4

Press **SAVE–F3**.

Now when the program runs, it will keep repeating as long as the run input is 'On'. This is an example of an Absolute Jump.

The Jump Event Condition screen (Conditional Jump) would be used just as the input and/or output event logic equation.

Thus, the previous example could be made to Jump only if an external input is 'On'.

Run Mode – Running the Program

Programs can be run from the operator interface unit (Hand–Held Terminal) or from digital I/O.

Running Programs from the Hand–Held Terminal

Press **'RUN'** to get the Run mode screen, then press **RUN–F1** key to get the following screen:

RUN AXIS[1]	QUIT	F1
	AXIS	F2
PROGRAM#[10]	<SEL	F3
	RUN	F4

Press **Axis–F2**, to get the correct axis and press **<SEL–F3** to get the profile to run.

CAUTION
1. Pressing **F4** now should cause motion. Be sure the machine or equipment is in proper configuration to operate. Assure that Enable and E-Stop circuits are active and operating correctly. 2. Selected profiles can be run from either the digital I/O or from the terminal. Pressing the **'Stop'** key will always STOP any motion running.

Press **RUN–F4** key: The selected axis should start running and the following screen will appear:

RUN AXIS[1]	QUIT	F1
	AXIS	F2
PROGRAM#[10]	<SEL	F3
RUNNING	STOP	F4

Pressing **Stop–F4** will stop the selected axis program motion.

The **'Stop'** key STOPS all motion – both axes.

Running Programs from Digital I/O

Programs can be commanded to run from the digital inputs in various ways:

1. If the program number has been selected to run with the Hand-Held Interface, then that program will run when the corresponding axis Run input is activated.

RUN AXIS[1]	QUIT	F1
	AXIS	F2
PROGRAM#[10]	<SEL	F3
	RUN	F4

Select the program to run by pressing **<SEL–F3**, until the desired program number is displayed. (Program must have been previously programmed).

2. The program to run can be selected from digital inputs by using the select inputs. Refer to Programming Reference Manual for use of this feature.

List

Several screens provide a function called List. Selecting this key will send a printer ready listing of the selected parameters through the RS232C port. For example, to print a listing of the axis gains, select **F4** from the following screen:

CONFIGURATION	↑ EDIT	F1
		F2
AXIS		F3
GAINS	↓ LIST	F4

Refer to SMC20 Installation section (page ##) for printer connections. The printer must be configured as follows:

Serial/Parallel:	**Serial**
Baud Rate:	**9600**
Paper Orientation:	**Portrait ***

If you have any problem, refer to Troubleshooting section (page ##).

***** Portrait is the default paper orientation for most printers. For example, this page is printed in portrait orientation. If the printer is set up for landscape orientation (where the page is printed lengthwise) some data may be missed.

Troubleshooting

Introduction

If you experience a problem with the SMC20, it will normally be associated with one of four conditions:

– Can't test the initial set–up

– Can't initiate motion

– Does not control motion well

– General failure

Through the use of the error bits described in the General Faults section (page ##) and helpful hints described in the following sections, an operator, in most instances, will be able to determine a solution to a problem. If, after following the troubleshooting procedure, the unit is suspected of failure, call Vickers and ask for a SMC20 service representative to verify your diagnosis and aid in resolution of the problem. Be sure when you call, to have the model number and serial number of the SMC20 in question.

Before performing troubleshooting procedures, be sure you are intimately familiar with the operation of the SMC20 and that a means to terminate unintended motion and electrical power to the unit exist and are easily accessible.

Also, be sure to turn 'Off' all power before attempting to remove the I/O terminal strips, feedback devices or communication devices (including the Hand–Held Terminal) from the SMC20.

When starting up a system, it is best to initialize the electronics first.

Can't Test Initial Set–Up

InTesting Initial Set-Up section (page ##), the operator is instructed to jog the SMC20 to initiate motion. If the axis does not move, it is most likely because the axis gains or axis parameters are incorrect. Another possibility is the actuator travels to an extreme limit in an uncontrolled manner. Another type of fault relates to the physical hardware.

Gains Not Set Properly

- Make sure the Kp gain is set to at least 1000. Kp is one of the axis gain parameters in the Configuration mode.
- Try the default Jog Velocity first. If the Jog Velocity is set too high and/or the Maximum Position Error is set too low, it is possible to trip the 'Maximum Position Error' (Error bit I/O addresses 235 or 243 set 'On' – see General Faults section, page ##).
- Any time a position error bit (General Faults section, page ##) is set 'On' by a fault, as described in some of the following descriptions, it is necessary to first reset the error (see section) using the Reset screen in the Run mode before attempting a new operation.
- Verify that the run inputs are not active.
- Verify that the Maximum Position Limits are not exceeded (Error bit I/O addresses 232, 233, 240, and 241 set 'On' – see General Faults section, page ##). It may be necessary to jog in the opposite direction of the limit in order to achieve motion.
- Verify Enable (ENBL) input is active.
- Verify the Enable Axis (EN*) input LED is 'On'. This input may be toggled after occurrence of an error to reset.

Uncontrolled Motion

- Loss of feedback while jogging will cause the axis to move to either extreme position out of control. Verify the position feedback reading in the Monitor mode. If there is no indication of position and the Enable (ENBL) is active, then the feedback device or the related cabling is faulty or the controller is set to open loop.
- If the polarity of the feedback signal is the same as the axis command signal, a 'run away' condition will occur. Power down and reverse the leads of the feedback device or the axis command signal, whichever is easiest.

Hardware Fault

- Verify fuse is not blown or loose. Replace with BUSS AGC 2 amp or equivalent. See Table 2–1 for value.
- If power supply LEDs are flashing, check all power connections including feedback device and communications wiring for a short or improper connection.
- If the CPU fault LED is 'On', contact a Vickers service representative.
- Verify that the Hand-Held Terminal is connected to the RS485 communication port.

Can't Initiate Motion

In this section, it is assumed that the SMC20 can be jogged, but motion controlled by the execution of the program cannot be initiated, or instantly stops and sets an error when initiated. If the unit cannot be jogged, refer to Section 5.2 before attempting the following procedure.

Whenever motion cannot be commanded, it is most likely due to conflicting operations that have been invoked or the controlling program is not selected correctly. The next likely possibility is the existence of conflicting axis parameters which allow the axis to start, but then instantly stop.

Conflicting Operations or Improper Program Selection

- Verify that the Enable Axis (EN*) LED is illuminated.
- Verify that the Axis Run (*RUN) LED is illuminated. It is possible to terminate motion using the STOP function on the Hand-Held Terminal and still have the Axis Run LED illuminated. Toggle the Axis Run input to ensure motion can once again be initiated by the program.
- Verify that the open loop mode has not been invoked in the Run mode.
- If using the Home function, ensure the Home Position has been established.

  ** = X or Y depending on which axis. (Axis 1 = X, Axis 2 = Y)*
- If invoking the program through the Program Select lines, be sure the right combination of select lines are high to select the intended program number. (Remember each line represents a weighted power of 2. For example, to run program 10, select line 2 and select line 8 must be high.)

Conflicting Parameters

- If the commanded velocity is too high and/or the Maximum Position Error is set too low, it is possible to trip the 'Maximum Position Error' (Error bit I/O address 235 or 243 set on – see Section 5.5). If this is the case, set the Maximum Position Error to a much larger value – reset the error (in Run mode), then tune the axis according to Appendix A. After the axis is tuned, reduce the Maximum Position Error to an acceptable number.
- It is possible to trip the 'Maximum Position Error' with a seemingly acceptable acceleration or velocity command if the axis gains are too low. Refer to Appendix A.

Motion Not Controlled Well

When dealing with erratic operation, in most instances it is due to stray signals from EMI or RFI sources entering the controller. Be sure all signals are shielded correctly and grounds are installed according to the recommendations described in SMC20 Installation section (page ##).

When dealing with unacceptable accuracies, repeatabilities, or long cycle times, and you are confident the components of the control system have been selected and installed correctly, it is most probable that greater attention must be paid to the tuning of the axes. Refer to Appendix A.

General Faults

The following addresses are dedicated to error tracking in the SMC20 and other versions utilizing the Hand-Held Terminal:

Address	Error Condition
232	Over travel + for Axis #2
233	Over travel – for Axis #2
234	Profile select error for Axis #2
235	Maximum position error for Axis #2
236	Profile segment error for Axis #2
237	Position sensor error for Axis #2
238	DSP error for Axis #2
239	Spare error bit for Axis #2
240	Over travel + for Axis #1
241	Over travel – for Axis #1
242	Profile select error for Axis #1
243	Maximum position error for Axis #1
244	Profile segment error for Axis #1
245	Position sensor error for Axis #1
246	DSP error for Axis #1
247	Spare error bit for Axis #1

You could assign one of these to an output, for example, to know immediately when that error has occurred.

Use the Monitor mode to view these addresses to see what caused the error.

The cause of error can be quickly viewed in the Monitor mode – Alarm screen:

```
AXIS[ 1] ALARMS      ↑       F1
FLAGS(00010000)              F2
           ↑            ←    F3
9MAX POS ERROR)      ↓  →    F4
```

In this example, the "1" represents a Maximum Position Error. Each "Bit" represents a possible error. 0 = No Error, 1 = Error. Use **F3**, **F4** to scroll across them to display a message on the error flags meaning.

List Function Inoperative

The List Function sends a printer ready listing of the selected parameters through the RS232C port.

If List is not working, use the following procedure to diagnose the problem.

1. Verify the printer cable is connected to the printer's serial port. Most printers have both serial and parallel ports.
2. Verify the printer is set up for 9600 baud.
   9600 baud, 8 data bits, 1 stop bit, no parity
3. Try switching pins 2 & 3 on **one** end of the cable.
4. Verify cable construction. Check for shorted pins and continuity. See Table 2-6. RX, TX, and COM are the only signals needed.

Setting the PID Gains

PID Parameters, Install Conditions

At startup, the **PID** gain settings are zero.

PID gains are AXIS GLOBAL VALUES which are set from the CFG mode – Axis Gains (see Axis Parameters section, page ##). Two PID gains can be altered in any segment by using the GAIN feature when programming the segment.

Open Loop Mode

If the loop has been deliberately opened using the screen available in Run mode, the **PID** algorithm is disabled. An offset in counts (range 0 to +/–8191 counts) can be entered which directly generates an output. The output produced will be:

(Maximum for the Selected Output) * (input counts)/8191

WARNING
Open loop operation of a normally closed loop can be hazardous.

Since the actuator will generally move at maximum velocity with full controller output and since an open loop can easily run to the maximum available position, it is suggested that a small offset be used and advance precautions should be taken to prevent injury or damage.

Closed Loop Mode, General Discussion

At each update interval, the difference between reference position and actual position is computed in digital form. This is the “Position Error”.

If only proportional gain (Kp) is used, constant SMC20 output corresponds to a constant position error. This required Position Error is called “Following Error” and is the amount by which measured position is behind the reference position for a given velocity.

The Position Error is routed through a **PID** digital compensation algorithm. The version implemented in this controller has a number of features specifically aimed at common electric servo loop applications.

FS	Means full scale controller output
EU	Means engineering units (counts)
Kp	Proportional gain
Ki	Integrator gain
Kd	Derivative gain (Velocity)
Integrator	The integrator in the PID compensation in each axis of this controller is a “part time” integrator. After the AXIS IN POSITION function has been TRUE for a time equal to DELAY, the remaining position error (within the INTEGRATOR BAND) is integrated. The integrator output is limited to a maximum of plus or minus the INTEGRATOR LIMIT. Adjustable parameters are: DELAY (SEC) BAND (EU) LIMIT (%FS)

Table A-1. PID Parameters

Figure A-1. Control Algorithm

The output of the **PID** module is loaded into a 14 bit Digital–to–Analog Converter (D/A) at each update (+/– 8191 counts). Any D/A offset programmed as an axis parameter (CFG mode) is added to the **PID** output in the D/A converter.

The Maximum Position Error which can be processed is limited. Note that Maximum Position Error programmed in CFG or PRG mode, sets up a value beyond which an error routine is triggered. It DOES NOT set the internal limits in the **PID** algorithm.

Jumper Settings

Factory Jumper Settings

The encoder interface card has jumper selectable ouput voltages for the encoder power. Refer to Figure B–1, Position Interface Card Encoder Jumpers.

Figure B-1. Position Interface Card Encoder Jumpers

The Main Board (CPU) has many factory set jumpers, no user configurable jumpers are provided on this board. The factory jumper positions are shown in FigureB-2 – SMC BD Jumpers.

Figure B-2. SMC Bd Jumpers

Index

Eaton Hydraulics
15151 Highway 5
Eden Prairie, MN 55344
Telephone: 612 937-7254
Fax: 612 937-7130
www.eatonhydraulics.com

46 New Lane, Havant
Hampshire PO9 2NB
England
Telephone: (44) 170-548-6451
Fax: (44) 170-548-7110

714 (Released 5/95)

Printed in U.S.A.