## TABLE OF CONTENTS

### CHAPTER 1. INTRODUCTION.

### CHAPTER 2. PROGRAMMER INSTALLATION

### CHAPTER 3. FRONT-PANEL PROGRAMMING

### CHAPTER 4. USING THE PROMSAVE ACCESSORY

## CHAPTER 5. REMOTE PROGRAMMING

## CHAPTER 6. MAINTENANCE AND CALIBRATION

## APPENDICES

## LIST OF FIGURES

## LIST OF TABLES

# Chapter 1. Introduction.

## 1.1. HOW TO USE THIS MANUAL.

This manual describes how to use the JC Systems Model 240A Dual-Channel Programmer.

### 1.1.1. How to Find Information.

**Table of Contents**

This reference aid lists major topics in the order they appear. It's an outline of the manual that also shows the page on which the discussion of each topic starts. All the figures and tables are listed separately at the end of the table of contents.

**Index**

The index is on the very last pages of the manual. It indexes each paragraph and subparagraph of the manual in alphabetical order.

**Reference Drawings**

Electrical schematics and assembly drawings of the programmer's principal components are grouped together at the back of the manual, in the last pages immediately before the index.

**Appendixes**

Appendix A contains a sample program worksheet. You can make as many copies of this as you need for your own use. Appendix B contains typical software commands used when interfaced with a remote computer. Appendix C contains instructions for installing and using the Model A2014 Multi-programmer Synchronizer option. Appendix D contains copies of schematics and assembly drawings for the programmer and related components.

### 1.1.2. How the Manual Is Organized.

The following summarizes the contents of this manual by chapter.

#### Chapter 1 - Introduction.

Summarizes the applications, components and features of the Model 240A Dual-Channel Programmer, usually referred to simply as "the programmer".

#### Chapter 2 - Installation

Describes installation of the programmer, including interfacing requirements and switch settings.

#### Chapter 3 - Front Panel Programming

Explains how to enter a program into the memory at the front panel keyboard . The programmer can then be used with JCS controllers to exercise direct digital control of process parameters. All features can be used in programmed operation. The entered program remains in memory until revised. A helpful feature: any parameter of any program can be changed at any time -- you don't have to reenter the whole program to change one parameter.

#### Chapter 4 - Program Storage with PromSave™

Tells how to store and retrieve one or more programs using the PromSave™. This JC Systems accessory uses removable, reusable EEPROM (electrically erasable, programmable, read-only memory) cartridges that hold the entire content of a Model 240A, 510, or 520 programmer's memory.

#### Chapter 5 - Remote (Computer) Operation

How to interface a remote computer with the programmer. Using the Model 240A's standard RS-232C serial interface or either of the interface options (RS-422A/485 or IEEE-488), the remote computer exercises complete control of programming functions.

#### Chapter 6 - Preventive Maintenance

The last chapter of the manual describes maintenance of the system.

### 1.1.3. Conventions Used in This Manual.

1. Names of pushbutton switches and displays are shown in ALL CAPITAL LETTERS. If the name appears on the equipment, spelling is exactly as shown there.

   *Example:* SELECT SETPNT

2. If an item is shown on a figure, the figure callout (item number) appears in parentheses after the item name is mentioned for the first time in each paragraph or step.

   *Example:* Press SELECT SETPNT button (Figure 1-1, Item 8).

   If the figure number does not appear with the item number, the item is on the last figure number referenced.

   *Example:* Refer to Figure 1-1 and proceed as follows.
   1. Press SELECT SETPNT button (8).

3. Standard abbreviations are not defined. However, the first time a non-standard abbreviation or acronym is used, its meaning is spelled out in parentheses.

   *Example:* PCB (printed circuit board).

## 1.2. SYSTEM DESCRIPTION.

The Model 240A is designed to program JC Systems stand-alone digital controllers used for direct digital control of temperature and related process values, such as pressure or humidity. The programmer comes equipped with a RS-232C serial interface. IEEE-488 GPIB or RS-422A/485 serial network communication interfaces are available as options.

This JCS product includes many features that result in easier use and more effective applications. These include random access to any step within the programmer's memory; internal nested looping capability in addition to repeating the complete program, soft start to prevent thermal stress to devices under test; switch-selectable deviation limits; a split-range current loop for reverse/direct output; long-life battery backup for program memory, and many other features. (See the product data sheet provided at the beginning of this manual for more information on features.)

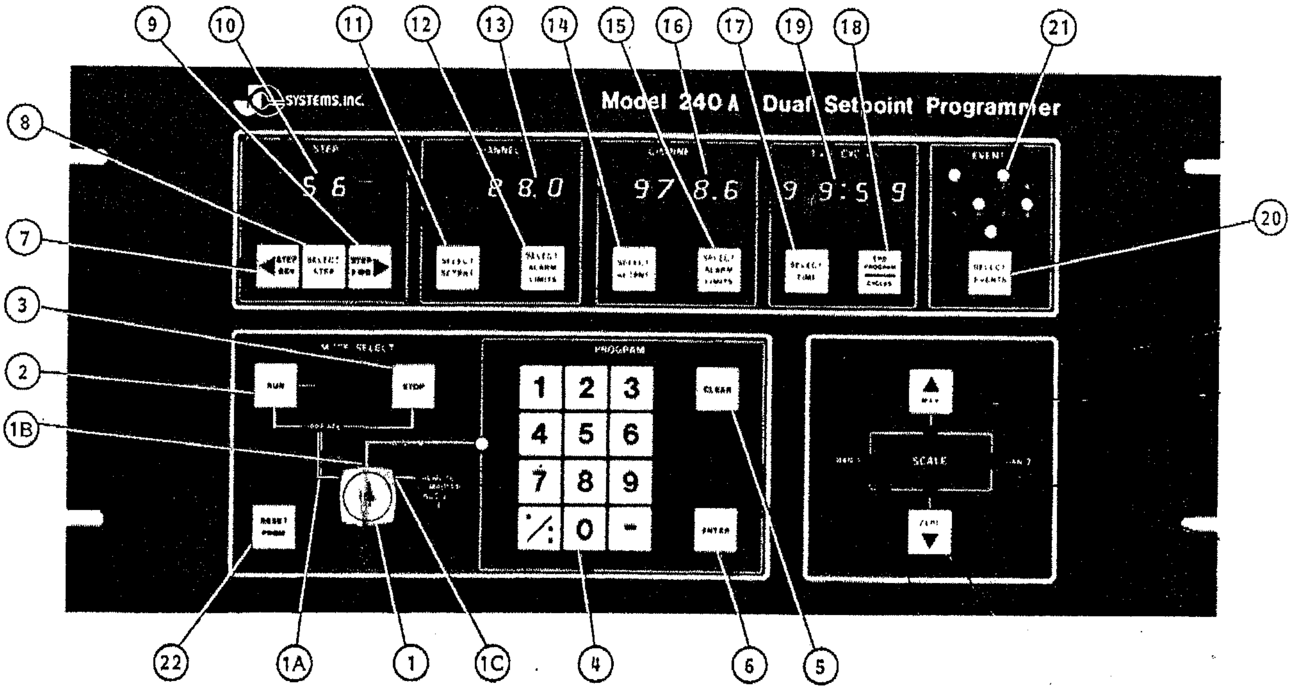


*Figure 1-1: Programmer Front Panel.*

## 1.4. PRINCIPAL COMPONENTS.

Principal components of the programmer are its front panel, which contains the operating controls; the rear panel, where electrical connections are made, and the printed circuit boards (PCB's) within the cabinet. Some switches, which must be set before beginning operation, are mounted on these PCB's. Procedures for making the electrical connections and setting switches are given in Chapter 2.

# Chapter 2. Programmer Installation.

## 2.1. INFORMATION PROVIDED.

This chapter describes unpacking and mounting the programmer, connecting electrical power, and setting DIP switches as required for desired operation.

## 2.2. UNPACKING.

Remove all protective packing and tiedowns from the Model 240A and remove the programmer from its shipping container.

## 2.3. MOUNTING.

### 2.3.1. Rack Mounting.

Mount the programmer in a standard 19-in. electronic rack, allowing at least 8-3/4 in. rack space to accommodate its height.

### 2.3.2. Bench Mounting.

For bench use, be sure to install rubber feet (available at no charge from JC Systems) on the bottom of the programmer to prevent damage to the benchtop.

## 2.4. CONNECT POWER AND CONTROLLERS.

1. Plug in power cable to 117V, 50/60 Hz grounded socket (Figure 2-1, Item 1).
2. Connect digital controller(s) to I/O ports J1 (2) and J2 (3).

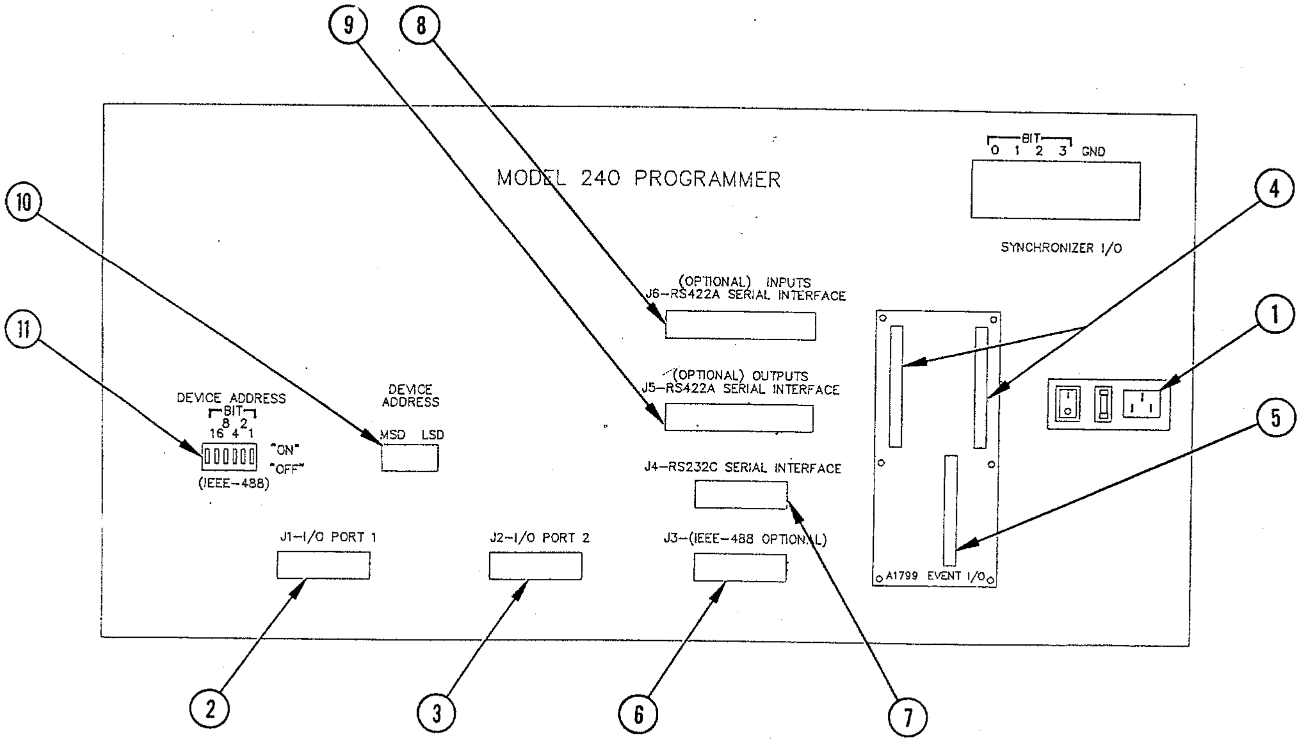


*Figure 2-1: Rear Panel Layout -- Standard Configuration*

## 2.5. CONNECT EVENT AND ALARM SOLID STATE RELAYS.

Connect alarms and events solid state relays (SSR) to appropriate terminals of TB1 (Figure 2-1, Item 4) on the A1799 Event I/O board as shown in Table 2-1. (If you're using the JCS A1748 Event Relay Board Accessory, you can connect the cable supplied with that accessory directly to panduit J1 on the I/O board (5); the pin assignments correspond with the terminals).

All SSR outputs are open collector logic type limited to 50 mA and 50 Vdc maximum. TB1-12, the internal +9V source, should be used as the positive source for photo-isolated solid-state relays (SSR) on the alarm output drives.

Terminal No. 10 is alarm #2 output; No. 11 is alarm #1 output, and No. 12 is the +9 Vdc unregulated internal voltage (100 mA max) used as source voltage for SSR. The terminals identified as 14 through 19 are used for external control signals; their use will be explained later.

## TABLE 2-1: TERMINAL ASSIGNMENTS FOR EVENTS CONNECTIONS TO SSR.

| TERM. NO. | FUNCTION | TERM. NO. | FUNCTION |
|---|---|---|---|
| 1 | Event 1 | 7 | Event 7 |
| 2 | Event 2 | 8 | Event 8 |
| 3 | Event 3 | 9 | Alarm/Event protection |
| 4 | Event 4 | 10 | Alarm #2 output |
| 5 | Event 5 | 11 | Alarm #1 output |
| 6 | Event 6 | 12 | + 9 V DC unregulated internal source (100 mA max) (SSR source voltage) |

## 2.6. SET DIP SWITCHES.

### 2.6.1. Set Front Panel Dip Switch Z33.

Certain operating characteristics of the programmer can be selected by changing the bit positions of DIP switch Z33 (Figure 2-2) on the inside of front panel display board A1840 as specified below. Note that the paragraph headings do not correspond to the actual legends on the PCB.

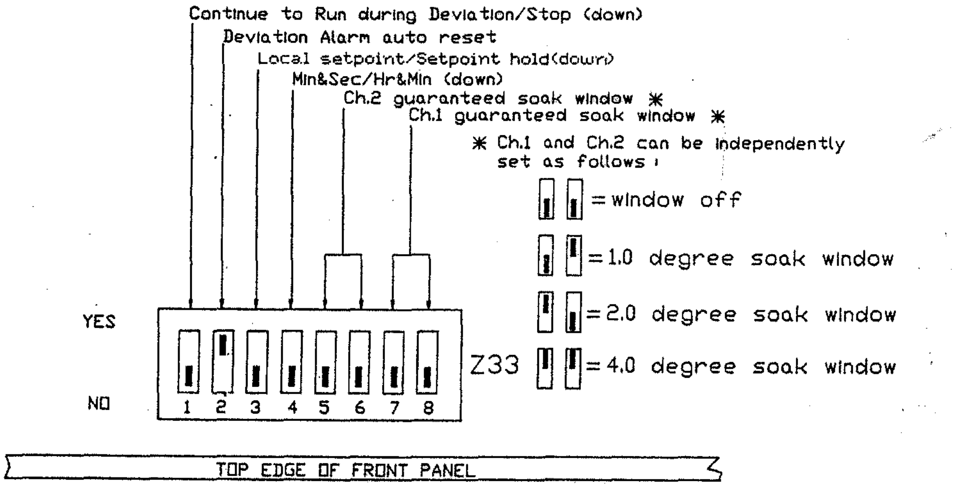

Continue to Run during Deviation/Stop (down)
Deviation Alarm auto reset
Local setpoint/Setpoint hold(down)
Min&Sec/Hr&Min (down)
Ch.2 guaranteed soak window ✱
Ch.1 guaranteed soak window ✱
✱ Ch.1 and Ch.2 can be independently set as follows :
= window off
= 1.0 degree soak window
= 2.0 degree soak window
= 4.0 degree soak window
YES
NO
1 2 3 4 5 6 7 8
Z33
TOP EDGE OF FRONT PANEL

*Figure 2-2: Z33 DIP Switch Settings.*

#### *2.6.1.1. Channel 1 Guaranteed Soak Window.*

Bit positions 7 and 8 of Switch Z33 can be set to select the soak window for Channel 1. The Channel 1 controller will wait (even though step duration has elapsed) until process variable No. 1 is within its setpoint ± the selected soak window before leaving this program step. Figure 2-2 shows how to set the switch to turn off the soak window or select soak windows of 1, 2, or 4 degrees. During this guaranteed soak window, the programmer will hold operation (colon in TIME/CYCLES display will stop flashing) until process variable No. 1 is within the selected limit.

*2.6.1.2. Channel 2 Guaranteed Soak Window.*

Bit positions 5 and 6 of Switch Z33 can be set to select the guaranteed soak window for the Channel 2 controller in the same manner as positions 7 and 8 are used for Channel 1 (see preceding paragraph and Figure 2-2).

*2.6.1.3. Set Time Base.*

With the MIN&SEC/HR&MIN, bit position 4, set to YES (MIN&SEC), time is measured in minutes and seconds from 0 seconds to 99 minutes, 59 seconds. In the NO (HR&MIN) position, time is measured in hours and minutes from 0 minutes to 99 hours and 59 minutes.

*2.6.1.4. Local Setpoint/Setpoint Hold.*

When the LOCAL SETPOINT/SETPOINT HOLD switch, bit position 3, is set to YES (LOCAL SETPOINT) position, program control reverts to the front panel of the controller. This allows the user to input a final setpoint after the program ends. When the switch is set to NO (SETPOINT HOLD) position, the setpoint specified at the end of the program (in program memory) will be used as the final setpoint.

*2.6.1.5. Alarm Auto Reset.*

Placing the DEVIATION ALARM AUTO RESET, bit position 2, to OFF (no AUTO RESET) causes the following: whenever alarm limits are exceeded, the programmer stops (TIME/CYCLES colon stops flashing), the alarm limit energizes, and the applicable SELECT ALARM LIMIT LED flashes. These conditions continue until parameters returns within limits and the RUN button is actuated (by manual or remote computer operation).

When the switch is set to YES (AUTO RESET), the programmer will automatically resume operation as soon as an out-of-limits condition has ended. At the same time, the alarm outputs are reset.

*2.6.1.6. Run During Deviation.*

When the CONTINUE TO RUN DURING DEVIATION, bit position 1, is set to ON, the programmer will continue running even though alarm limits have been exceeded. The alarm outputs will either latch (continue) or automatically reset (turn off), depending on ALARM AUTO RESET bit 2 position.

### 2.6.2. Programmer Control Logic Board Switch S1.

This switch (Figure 2-3) is used to select single-step operation and to enable the multiple programmer synchronizer option as explained below. It is also used to select serial or IEEE-488 interface mode, which is discussed in Chapter 5.

## TABLE 2-2. CONTROL LOGIC BOARD DIP SWITCH S1 SETTINGS

| S1 BIT | FUNCTION | SETTING | SELECTS |
|---|---|---|---|
| 1 | Interface Select | ON (Serial Mode) | RS-232 or RS-422/485 serial interface |
| | | OFF (GPIB Mode) | IEEE-488 parallel interface |
| 2 & 3 | Not used on 240A | | |
| 4 | Single Step | ON | Program stops at end of step if Event 6 ON |
| | | OFF | Ignores Event 6 status |
| 5 & 8 | Not used on 240A | | |
| 6 | Manual restart from front panel after single-step stop | ON | Can restart with front panel RUN button while in remote computer mode after single-step stop |
| | | OFF | Front panel RUN button won't work in remote computer mode |
| 7 | Multiple programmer synchronizer | ON | Enables optional synchronizer board (if installed) |

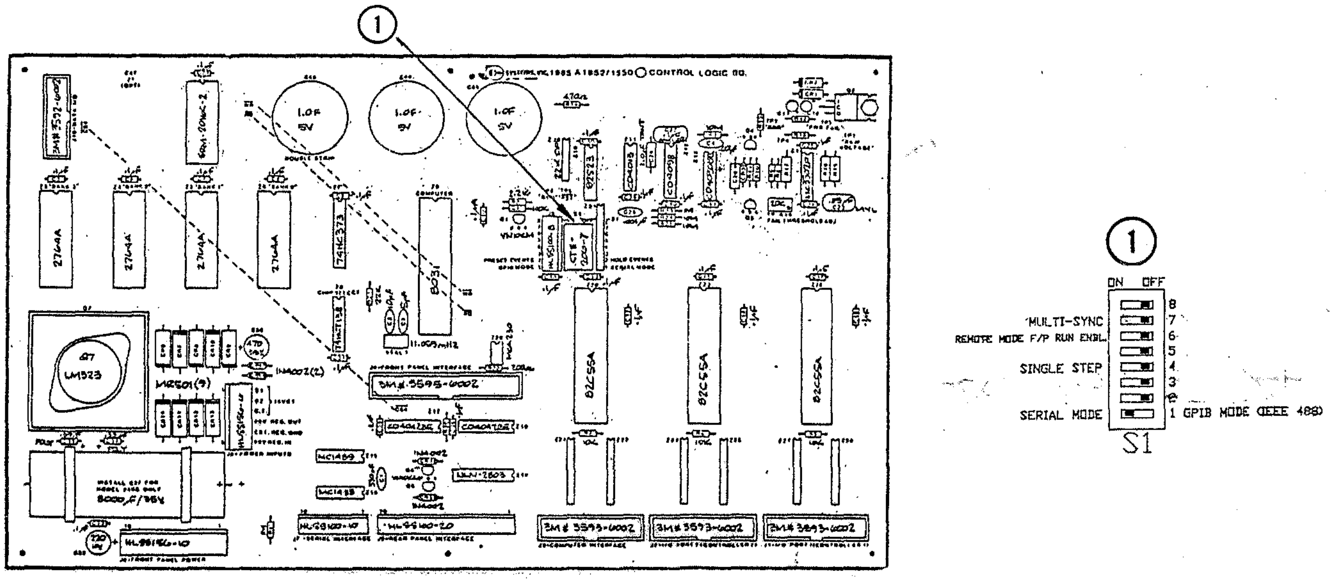


*Figure 2-3: Control Logic Board DIP Switch S1.*

### *2.6.2.1. Single-Step (Programmed Stop) Operation.*

Single-step programmed stop operation allows you to automatically stop a program at the end of a selected step. When Switch S1 Bit 4 is ON (set to Single Step) and Event 6 is enabled in a program step, program execution will halt at the end of that step. The STEP display will show the next step number. The time/cycle display will show the time for the next step, and the clock will stop. The last setpoint output will be displayed, latched, and maintained from the step where Event 6 was enabled. The event outputs will also remain in the configuration selected for the step in which Event 6 was turned on.

For example, assume that Step 2 of your program is a 2-hour soak at 100°C with Event 6 enabled and Switch S1-4 ON to invoke the program stop (single-step) feature. Step 3 is a 1-hour ramp to -55°C. At the end of the 2-hour soak in Step 2, the programmer will halt program execution. The setpoint will be displayed, latched and maintained at 100°C. The step display will show 3, the time display will show 1 hour, and the event display will show Event 6 on.

To restart program execution, either press the RUN button on the front panel (see next paragraph), activate the remote RUN switch (if installed), or use the remote computer RUN command.

### *2.6.2.2. Remote Mode Front Panel RUN Enable.*

When Switch S1 bit 6 is set to ON, you can manually restart a program in computer remote mode using the front panel RUN button after the single-step feature (Para. 2.6.2.1) has been used to stop the program. (The remote computer RUN command can also be used for restart.) If S1-6 is OFF, the front panel RUN button does not work in remote computer mode after a single-step stop.

### *2.6.2.3. Multiple Programmer Synchronizer Enable.*

Switch S1 bit 7 set to ON enables the optional synchronizer board (if installed). One or more programmers so configured will operate synchronously when connected to each other via twisted-pair cables to terminal blocks on the back of the programmers. (If a programmer is equipped with a synchronizer board but the synchro cable isn't connected, the programmer functions as a stand-alone unit regardless of the S1-7 setting.) Programmers equipped with the synchro option are shipped from the factory with S1-7 set to ON.

# Chapter 3. Front-Panel Programming.

## 3.1. INFORMATION PROVIDED.

This chapter describes how to plan and record a program using the worksheet, and how to enter the program in memory.

## 3.2. ENTER A PROGRAM.

### 3.2.1. Use the Worksheet to Plan a Program.

Figure 3-1 shows a typical profile that demonstrates the basic features of the programmer. The program assumes that the programmer is set for operation in °C with a time base in minutes and seconds.

You will find it much easier to enter a program if you first use a worksheet to write down the program. This way, you can spot potential problems and correct them without backtracking through the programmer displays. Figure 3-2 shows the profile from Figure 3-1 entered on a sample program worksheet.

In Paragraph 3.4, we will revise the program to introduce more advanced features. But for now, let's go through the program on Figures 3-1 and 3-2 step by step.

Step 1. Ramp from ambient to -50° in 15 minutes, 10° alarm deviation limit set. The programmer's soft start feature permits starting the program immediately from ambient. If a guaranteed ramp is desired, add a preliminary step to achieve the desired starting temperature.

Step 2. Soak at -50° for 15 minutes; alarm limits unchanged, no events turned on.

Step 3. Ramp to 100° in 30 minutes; alarm limits unchanged, no events on.

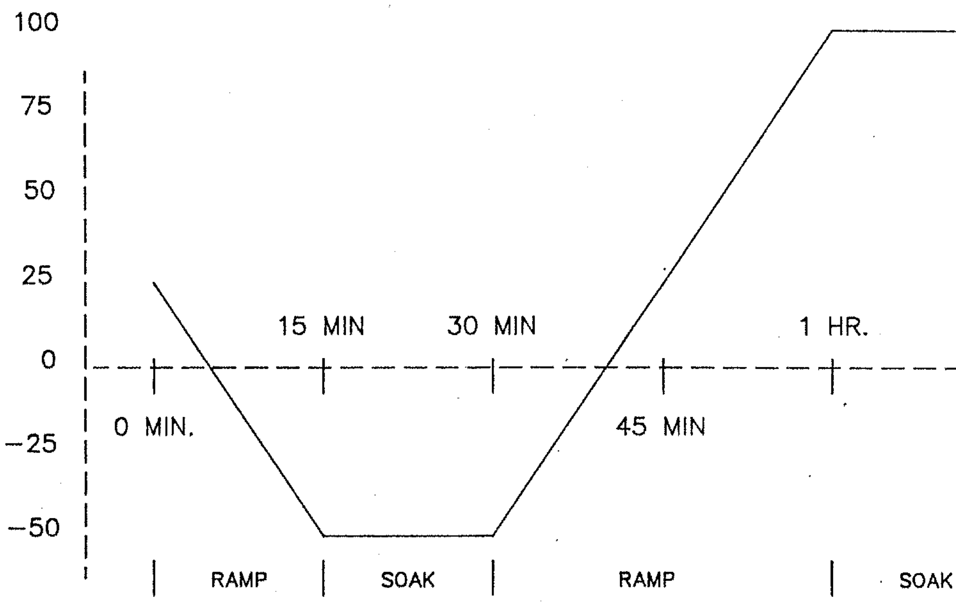
Step 4. Soak at 100° for 15 minutes; alarm limits unchanged, no events on.

Step 5. Ramp to 25° (ambient) in 15 minutes. Alarm limits unchanged, no events on.

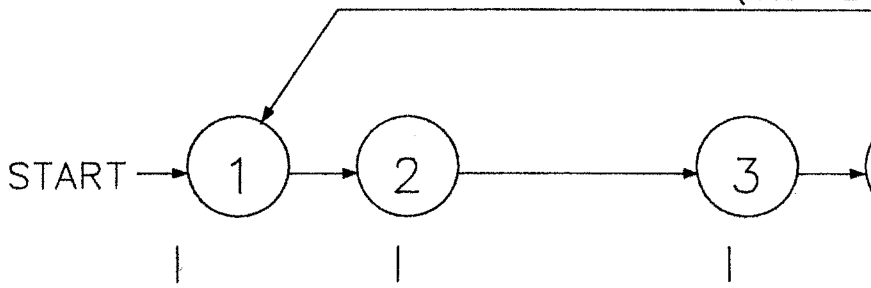
Step 6. Execute Program 5 Times and End Program. Note the 5 in the time column. This is a reminder to enter a 5 at the END PRGM/CYCLES step, so that the entire program will execute 5 times. (More about that when we actually enter the program.)

N PROGRAM STEP – "N" IS THE STEP NU

EOP N END OF PROGRAM STEP – STORES: Nu Mo



CYCLES = 5 (EXECUTE



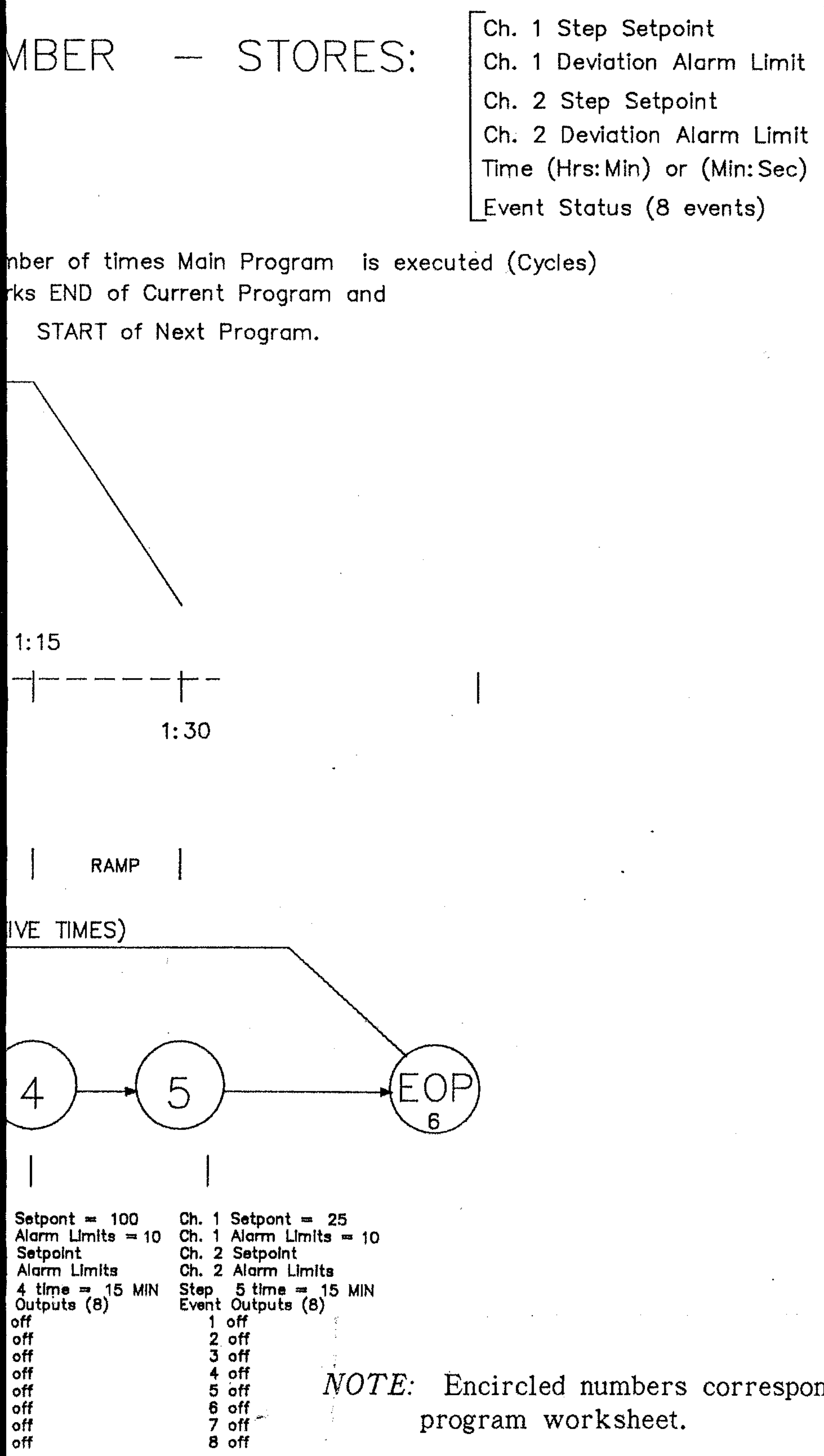
Ch. 1 Setpont = –50
Ch. 1 Alarm Limits = 10
Ch. 2 Setpoint
Ch. 2 Alarm Limits
Step 1 time = 15 MIN
Event Outputs (8)
1 off
2 off
3 off
4 off
5 off
6 off
7 off
8 off

Ch. 1 Setpont = –50
Ch. 1 Alarm Limits = 10
Ch. 2 Setpoint
Ch. 2 Alarm Limits
Step 2 time = 15 MIN
Event Outputs (8)
1 off
2 off
3 off
4 off
5 off
6 off
7 off
8 off

Ch. 1 Setpont = 100
Ch. 1 Alarm Limits = 10
Ch. 2 Setpoint
Ch. 2 Alarm Limits
Step 3 time = 30 MIN
Event Outputs (8)
1 off
2 off
3 off
4 off
5 off
6 off
7 off
8 off

Ch.
Ch.
Ch.
Ch.
Step
Even

LOOPS_2 4/1/86



*NOTE:* Encircled numbers correspond to step numbers shown on sample program worksheet.

*Figure 3-1: Profile for Program Listed on Sample Worksheet (Figure 3-2).*

## PROGRAM WORKSHEET

JC SYSTEMS INC. PROGRAM NO. Sample #1 (for Fig. 3-1) DATE: ____________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | | | | | | | | | | | | | |
| 1 | -50 | 10 | -- | -- | 15 | | | | | | | | |
| 2 | -50 | 10 | -- | -- | 15 | | | | | | | | |
| 3 | 100 | 10 | -- | -- | 30 | | | | | | | | |
| 4 | 100 | 10 | -- | -- | 15 | | | | | | | | |
| 5 | 25 | 10 | -- | -- | 15 | | | | | | | | |
| 6 | EOP | | | | 5 | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | |

*NOTES:*

1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

*Figure 3-2: Program Worksheet with Program Entered.*

### 3.2.2. Watch Program Length.

*3.2.2.1. Program Length and Programmer Storage Capacity.*

One complete program consists of all steps containing parameters and the EOP step. A program with nine steps is actually ten steps long (counting the EOP). Since the programmer can store a maximum of 91 steps, it has room for nine programs each containing ten steps total (9 steps plus the EOP step).

*3.2.2.2. Importance of End-of-Program Step.*

The EOP has the following functions:

1. It marks the end of one program and the beginning of another.
2. It specifies the number of times the program will be executed.

The first step of a program must be immediately preceded by an EOP step, unless the program starts at Step 1. If a program does not start after an EOP, the programmer will skip over channels and will not accept your input.

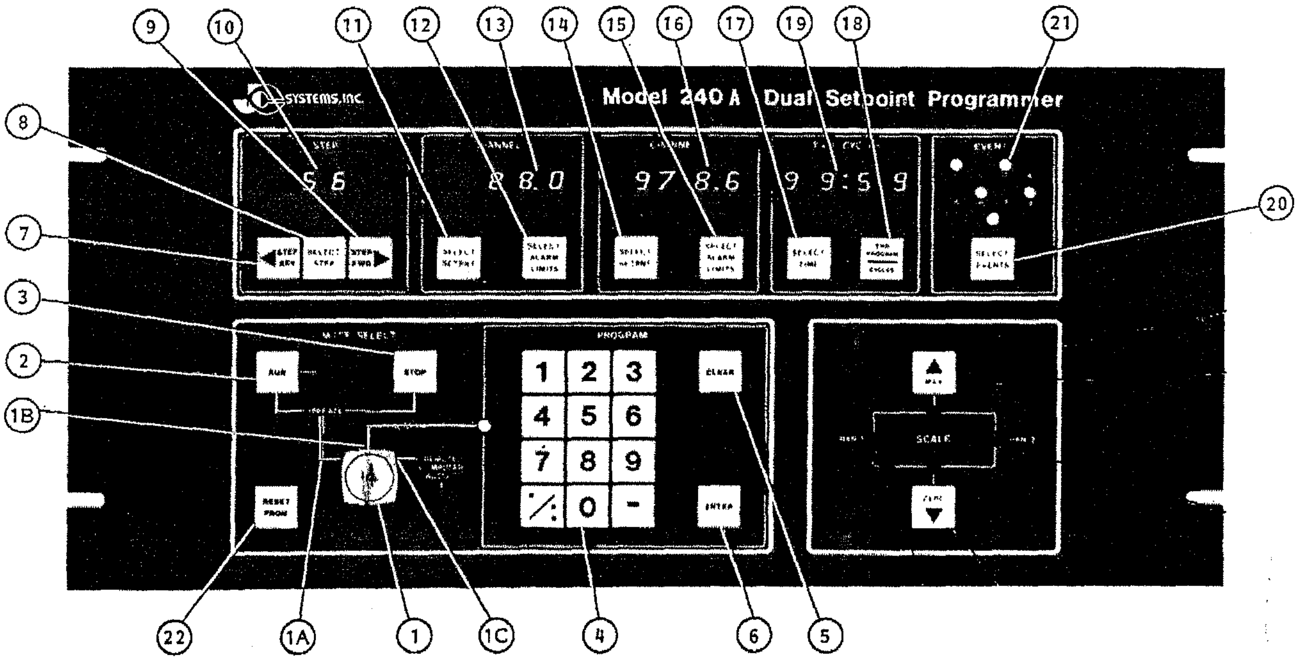
Hint: it's not necessary, and may even be inconvenient, to start the next program on the next available step after an EOP. Leave a few steps blank in case you want to revise and expand the program later. (Be sure to terminate the blank steps with an EOP step.)

## 3.3. ENTER THE PROGRAM AT FRONT PANEL.

Now that we've planned the program, let's enter it. For this first program, we will be using Channel 1 only. If you compare Figure 3-2 to the front display panel (Figure 3-3), you will see that the worksheet provides a place to enter each of the values you will program. The worksheet is arranged so that the values appear in the same order that the programmer pointer (the LED that lights above the SELECT buttons) follows as it moves across the top row of the programmer.

### 3.3.1. General Instructions.

1. The controller(s) being programmed must be in program mode (PROG indicator LED on), or the values will not entered in memory.
2. Remember that the CLEAR button (Figure 3-3, Item 5) can be used at any time to remove incorrect input, and the ENTER button (6) must be used to store input in memory. If the value already in memory is correct, simply press the ENTER button to keep that value.



*NOTE:* For convenient reference, a duplicate of Figure 3-3 is printed on the last page of this chapter so that you can see the figure when that page is folded out.

*Figure 3-3: Front Panel Buttons, Displays and Indicators.*

3. As you enter the program, the LED above the top row of buttons serve as a pointer to tell you what you are entering. For example, the pointer above the SELECT SETPNT button will light as soon as you select a program starting step. The SELECT SETPNT pointer will go out as soon as you have entered the setpoint, and the pointer above the SELECT ALARM LIMITS button will light.
4. The sample program lists the applicable value from Figures 3-1 and 3-2 in brackets [ ]. You could enter any similar program simply by substituting your own values.

### 3.3.2. Finding Switches and Displays.

Figure 3-3 shows the location of and identifies front panel displays and buttons you will use to enter the program.

### 3.3.3. Select Starting Step.

1. Insert key in keyswitch (1) and turn to PROGRAM mode (1B). PROGRAM LED lights and keypad becomes active.
2. If a program has been entered, there are two convenient ways to access the first available step.

   a. If you are curren ly on any step of the program you wish to run, press the RESET PRGM BUTTON (22).
   b. If you know the first step number of the program you want to run, use the random access feature to select the starting step as follows:
      1) Press SELECT STEP button (8). The SELECT STEP LED will light.
      2) Use the keypad (4) to input the desired step number, then press the ENTER button (6). SELECT STEP pointer will go out and SELECT SETPNT pointer will light. (You are ready to start programming.)

*NOTE*

The easiest way to keep track of available steps and programs is by using a program worksheet like Figure 3-2 as a log. (You can copy the blank worksheet in Appendix A for your own use.)

### 3.3.4. Enter Step 1.

*3.3.4.1. Enter Channel 1 Setpoint.*

The programmer will only run with a setpoint within the operating range of your system's controllers (as listed on the rear panel and the Calibration Sheet included at the front of this manual). To enter the setpoint, proceed as follows.

1. Using keypad (4), input desired value [-50°]. CHANNEL 1 display (13) will show inputted value.
2. Press ENTER button (6) to record value in program memory and advance program pointer to next parameter.

*3.3.4.2. Enter Channel 1 Alarm Limits.*

Enter a deviation limit of ±10° for Step 1 as follows.

*NOTE*

> If no alarm limits are desired, use the CLEAR button to make the entry. **Do not use** 0.0, which will place the programmer in a constant alarm condition.

1. Using keypad (4), input desired deviation limit value (10°). CHANNEL 1 display (13) will show inputted value.
2. Press ENTER button (6) to record value in program memory and advance program pointer to next parameter.

*3.3.4.3. Enter Channel 2 Setpoint.*

Channel 2 is not used for this program, so when the Channel 2 SELECT SETPNT pointer lights, press CLEAR (5) and then ENTER (6) buttons to advance pointer to Channel 2 SELECT ALARM LIMITS. This will clear Channel 2 setpoint for all remaining steps.

If Channel 2 *were* used, the procedure would be the same as that for Channel 1 (except that the values might be different).

*3.3.4.4. Enter Channel 2 Alarm Limits.*

Channel 2 is not be used for this program, so when the Channel 2 ALARM LIMITS pointer lights, press CLEAR (5) and then ENTER (6) buttons to advance pointer to SELECT TIME button (17). Since the Channel 2 setpoint was already cleared, this will clear Channel 2 and the program pointer will skip Channel 2 for all remaining steps.

If Channel 2 *were* used, the procedure would be the same as that for Channel 1.

*3.3.4.5. Enter Time (Step Duration).*

1. Using keypad (4), input desired value [15 minutes]. (The programmer will accept values from 0 minutes thru 99 hours, 59 minutes, or from 0 seconds thru 99 minutes, 59 seconds, depending on the setting of switch Z-33 bit 4 (time base setting.) Normally, a colon must be entered to select seconds (or minutes, if the hour/minute mode has been selected). In this case only, a period can be used instead of a colon. TIME/CYCLES display (19) will show inputted value.
2. Press ENTER button (6) to record value in program memory and advance program pointer to next parameter.

*3.3.4.6. Select Events.*

There are no events for this step (or for any step of this program; programming of events will be explained later). Press CLEAR (5) and ENTER (2) buttons; STEP display (10) will advance to next step (Step 2) and pointer will advance to channel 1 SELECT SETPNT.

### 3.3.5. Enter Additional Intermediate Steps.

For each additional intermediate step in the program (up to the final end-of-program step), the procedure is the same as it was for Step 1. Only the parameters change as shown in Figure 3-1 and repeated here for reference.

Step 2. Soak at -50 ±10° for 15 minutes.
Step 3. Ramp to 100 ±10° in 30 minutes.
Step 4. Soak at 100 ±10° for 15 minutes.
Step 5. Ramp to 25 ±10° in 15 minutes.

### 3.3.6. Enter End-of-Program (EOP) Step.

After all parameters for all steps of the program have been stored in memory, Step 6 will be displayed and the program pointer LED will be on over the Channel 1 SELECT SETPNT button (11). Enter the End of Program (EOP) step as specified below.

*3.3.6.1. Program Cycles.*

The CYCLES selection contained in the EOP step makes it possible to repeat the entire program as many times as desired. (One cycle equals one complete program execution.) Our sample program executes the complete program's heat/cool cycle five times. You can program as many as 9999 cycles with the EOP step. The cycle will execute the entire program as many times as necessary without using up valuable program memory.

1. To enter an EOP step with multiple executions (cycles), press END PRGM/CYCLES button (18). Program pointer LED over button will light and display will show current setting.
2. Using keypad (4), input desired number of program executions [5 for our sample.] TIME/CYCLES display (19) will show inputted value.
3. Press ENTER button (6) to record value in program memory. Program pointer automatically moves to first step of program just entered (Step 1), ready to begin operation.

*3.3.6.2. End Program After One Execution.*

1. To enter an EOP step for a program you only want to execute once, press END PRGM/CYCLES button (18). Program pointer LED over button will light and display will show current setting.
2. Using keypad (4), input desired number of program executions [1 if only one cycle is desired].
3. Press ENTER (6) button to record value in program memory. Program pointer automatically moves to first step of program just entered, ready to begin operation.

## 3.4. ENTER A PROGRAM WITH NESTED LOOPS.

### 3.4.1. Using the Nested Loop Capability.

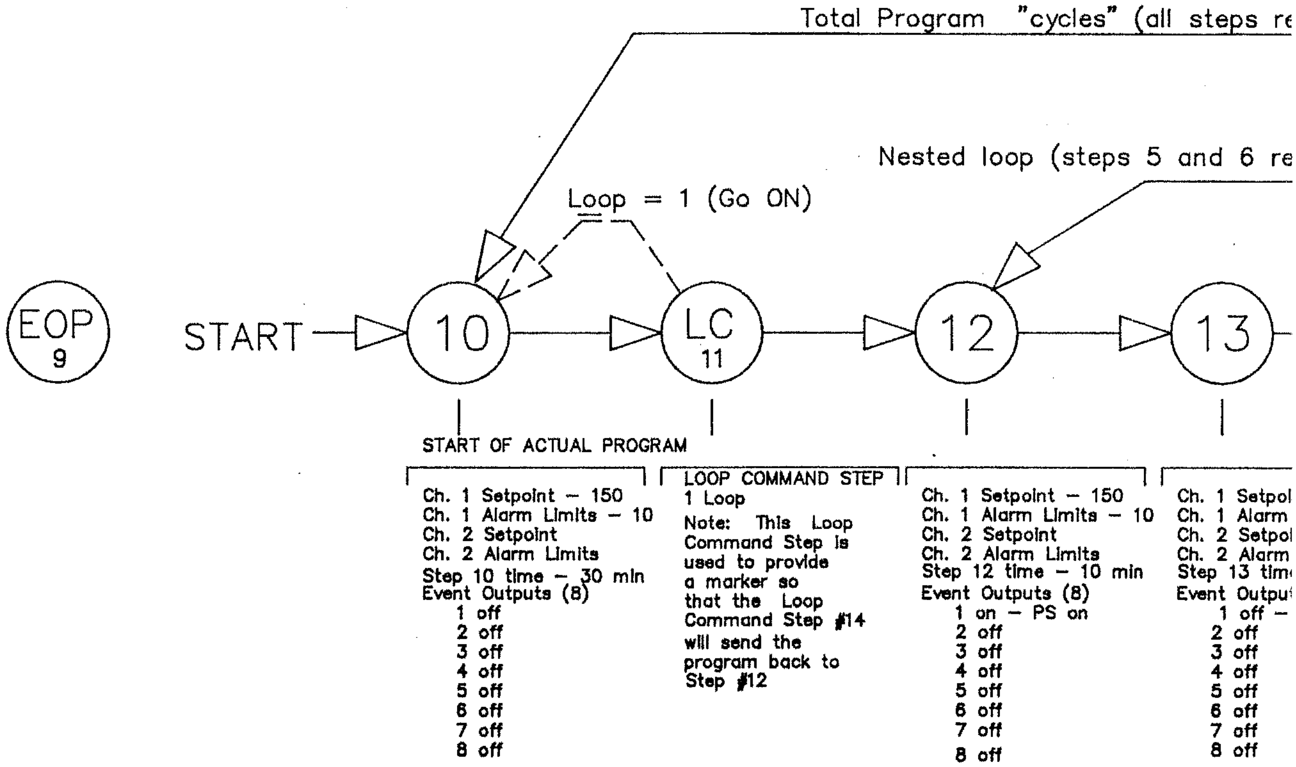
With the first sample program above, you saw how you could use the EOP step to cycle the entire program several times. The nested loop capability enables you to do the same thing with selected portions of the program.

In our second sample program, starting at Step 10, we've added such a loop. Now we're going to turn the power supply on and off 50 times within a program using a nested loop, *and* we're going to repeat the entire program 20 times. That means we're actually going to repeat Steps 12 and 13 a total of 1,000 times. Figure 3-4 graphically presents the program and its profile; Figure 3-5 is the worksheet for the program shown in 3-4.

N PROGRAM STEP – "N" IS THE STEP NUMBEF

LC N LOOP COMMAND STEP – STORES: Number of times (
Nested steps a
Marks END of Nes
Next section of

EOP N END OF PROGRAM STEP – STORES: Number of
Marks END
START



Total Program "cycles" (all steps re
Nested loop (steps 5 and 6 re
Loop = 1 (Go ON)
EOP 9
START
10
LC 11
12
13
START OF ACTUAL PROGRAM

Ch. 1 Setpoint – 150
Ch. 1 Alarm Limits – 10
Ch. 2 Setpoint
Ch. 2 Alarm Limits
Step 10 time – 30 min
Event Outputs (8)
1 off
2 off
3 off
4 off
5 off
6 off
7 off
8 off

LOOP COMMAND STEP
1 Loop
Note: This Loop Command Step is used to provide a marker so that the Loop Command Step #14 will send the program back to Step #12

Ch. 1 Setpoint – 150
Ch. 1 Alarm Limits – 10
Ch. 2 Setpoint
Ch. 2 Alarm Limits
Step 12 time – 10 min
Event Outputs (8)
1 on – PS on
2 off
3 off
4 off
5 off
6 off
7 off
8 off

Ch. 1 Setpoi
Ch. 1 Alarm
Ch. 2 Setpoi
Ch. 2 Alarm
Step 13 tim
Event Outpu
1 off –
2 off
3 off
4 off
5 off
6 off
7 off
8 off

Note: This example is to demonstrate the of the nested loop. The power supplies hc to be turned on for 10 min. and off for 10 min. 50 times.

Channel 2 is ignored in this example

LOOPS_1

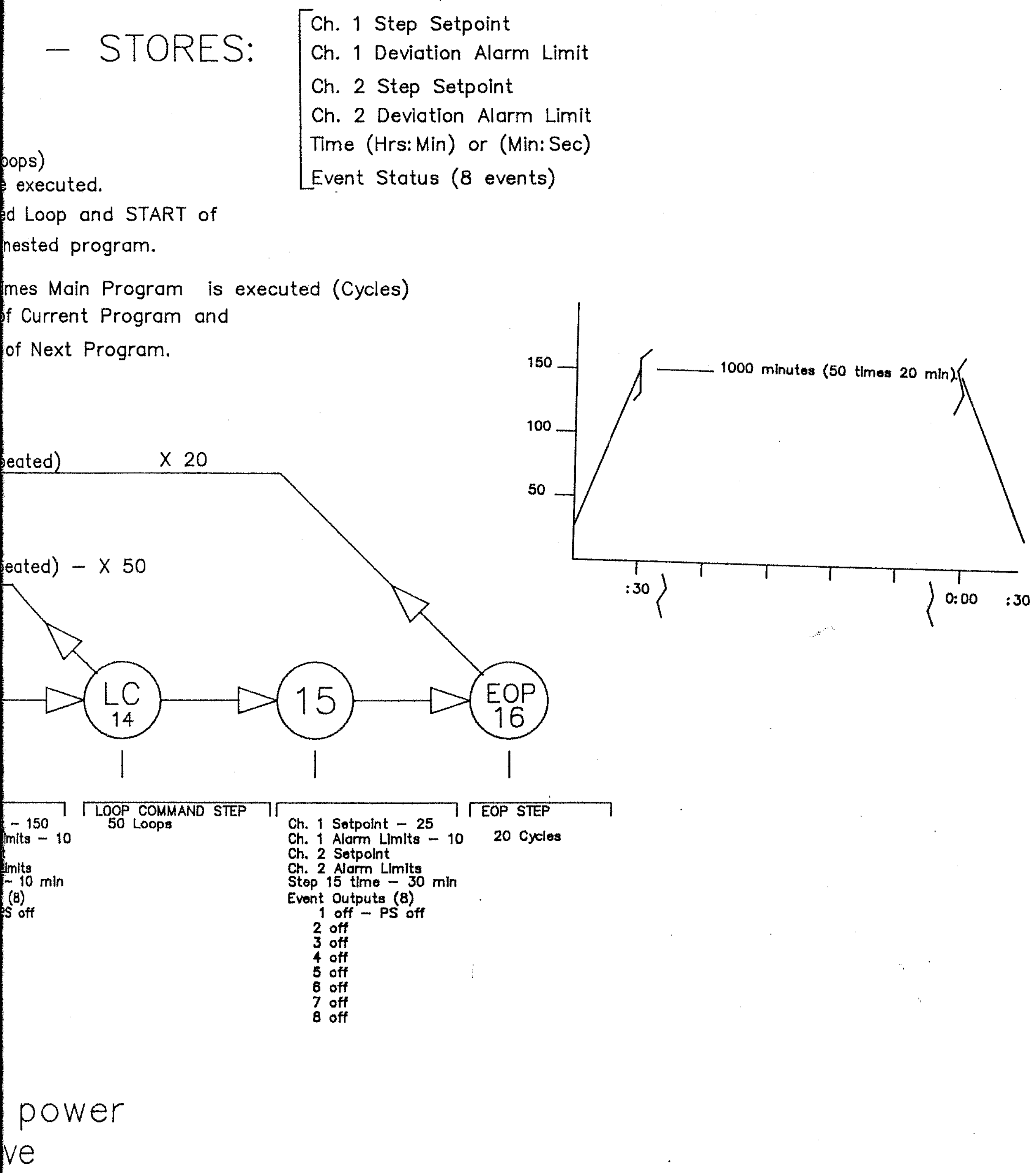


*Figure 3-4: Nested Loop Flow Chart and Profile.*

PROGRAM WORKSHEET

JC SYSTEMS INC.  PROGRAM NO. Sample #2 (for Fig. 3-4)  DATE: ____________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| * 9 | EOP | | | | 1 | | | | | | | | |
| 1Ø | 15Ø | 1Ø | -- | -- | 3Ø | | | | | | | | |
| 11 | LOOP | | | | 1 | | | | | | | | |
| 12 | 15Ø | 1Ø | -- | -- | 1Ø | ON | | | | | | | |
| 13 | 15Ø | 1Ø | -- | -- | 1Ø | | | | | | | | |
| 14 | LOOP | | | | 5Ø | | | | | | | | |
| 15 | 25 | 1Ø | -- | -- | 3Ø | | | | | | | | |
| 16 | EOP | | | | 2Ø | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 2Ø | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | |
| 3Ø | | | | | | | | | | | | | |

*This EOP step used to mark start of program (on Step 1Ø).

@font MX9I;NOTES: 1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

*Figure 3-5: Worksheet for Program Sample with Nested Loop.*

### 3.4.2. Programming a Nested Loop.

The sample program shown in Figures 3-4 and 3-5 incorporates two more programmer features: the nested loop and events programming. The following paragraphs explain these procedures.

*3.4.2.1. Loop Location.*

The programmer will not start on any loop command step. If the RUN button (3a) is pressed while in such a step, the RUN LED will flash and the programmer will return to the STOP mode.

*3.4.2.2. Loop Execution.*

A nested loop must contain its program steps in sequence. The beginning of the nested loop is marked either by an EOP step or by a loop command step. The loop will return to first step after that beginning marker and execute the program steps within the nested loop until it executes the number of loops specified in the end loop command step. Then the program exits the loop and proceeds to the step immediately following the end loop command.

*3.4.2.3. Loop Storage.*

Each loop command uses one program step, just like the EOP command. Up to the 91-step capacity of the program, a program can contain as many *sequential* nested loops as required. (Nested loops may not be nested *within* nested loops.)

*3.4.2.4. Loop Format.*

The nested loop consists of:

1. A marker at the beginning of the loop. This can be either a loop command or an EOP. No marker is required before Step 1.
2. The program steps to be performed during the nested loop.
3. A LOOP command step to mark the end of the loop and specify the number of times it will be executed.

When the programmer encounters a loop command step, it decrements the loop counter (number of cycles selected for the loop) by 1 and checks the count. If the count is not 0, the programmer moves backward linearly through the steps until it encounters either a previous loop command or an EOP command, then resumes program execution. If the count is 0, the programmer advances to the next programmed step.

To enter a nested loop that begins with the Step 1 of the program, you do not need a loop command step as a beginning marker. The beginning marker is needed only when the nested loop does not include the first step of the program, as is the case for our second sample.

### *3.4.2.5. Programming Procedure for a Nested Loop.*

For the example, we are entering the nested loop shown in Figures 3-4 and 3-5. The steps of this program are numbered 10 thru 16 to distinguish them from the first sample program.

1. Make sure there's an EOP step immediately before the first step of your program. The easiest way to do this is to enter an EOP step.
2. Start the program in the usual way, by entering desired program parameters for Step 10, the first step [Channel 1 setpoint 150°, alarm limits ±10, time 30 minutes, no events]. Advance to the next step.
3. For Step 11, mark the beginning of the nested loop by pressing the END PRGM/CYCLES button (18). A number between 0 and 9999 will appear in the TIME/CYCLES display (19). This is the value (if any) previously entered for that step.
4. Use the keypad (4) to input 1 (the beginning loop step is only executed once). Then press the END PRGM/CYCLES button to mark the beginning of the loop. **DO NOT PRESS THE ENTER KEY**, since the programmer would then interpret the entry as an EOP command instead of a loop marker.
5. Enter the steps you want executed in the nested loop (No. 12 and 13 of our sample program). The setpoint and alarm limits for both are the same as program Step 10, but the time changes to 10 minutes. We also want to turn on an event [#1, which energizes the power supply of the device under test] during Step 12, then turn it off during Step 13. To program these steps, proceed as follows.
   a. Enter the setpoint, alarm limit, and time for Step 12 using the same procedures previously described. The program pointer advances to SELECT EVENTS.
   b. Enable Event #1 for Step 12 using the procedure described in Para. 3.4.2.6 below.
   c. Repeat procedural Step 3.a above for program Step 13, then disable Event #1 using the procedure described in Para. 3.4.2.6.
6. At Step 14 of the sample program, press the END PRGM/CYCLES button (18). A number between 0 and 9999 will appear in the TIME/CYCLES display (19). This is the value (if any) previously entered for that step.
7. Use the keypad (4) to input the number of times you want the loop to execute [50 for our sample]. Then press the END PRGM/CYCLES button again to store the loop command. **DO NOT PRESS THE ENTER KEY**, since the programmer would then interpret the entry as an EOP command instead of a LOOP command.

8. Enter any additional operational steps (Step 15 in our sample).
9. Enter the EOP step. The nested loop program has an EOP step with 20 repeat cycles instead of the first sample program's 5 cycles, but the procedure for step entry is the same.

*3.4.2.6. Select EVENTS Status.*

The buttons used to clear or select events are as follows.

1. The first time after events are selected for a step, pressing the keypad (4) button corresponding to the event number toggles the selected event on and toggles off any previously selected events. (The corresponding event status LED (21) will toggle on and off as applicable.)
2. At any other time, pressing the keypad (4) button corresponding to the event number toggles the selected event on and off as indicated by the corresponding event status LED (21).
3. Pressing the CLEAR button (5) turns off all events.
4. Pressing the ENTER button (6) enters the current events status in program memory.
5. Once events status has been selected and the ENTER button pressed, STEP display (10) will advance to next step and program pointer will light over Channel 1 SELECT SETPNT button (11).

## 3.5. STORE THE PROGRAM.

No special procedure is required to store the program in the programmer's internal memory. The program will remain in memory until you either revise it or write over it. (The battery backup's minimum five-year life will protect the program almost indefinitely, even with repeated shutdowns or power outages.)

You can also save the program to a remote storage device and retrieve it later. See Chapter 4 for instructions on using the JC Systems PromSave™ accessory, or Chapter 5 for saving programs to a remote computer.

## 3.6. REVISE THE PROGRAM.

You can revise any step of a program without having to redo the entire program. You also can change one or more values within the step without affecting values you don't want to change. However, if you add steps to a program, you must re-enter any subsequent steps. The procedures for accessing and revising the step are given below. Numbers in parentheses refer to Figure 3-3.

### 3.6.1. Access a Step in Program Mode.

You can access the step(s) you want to revise by either random access or sequential selection.

*3.6.1.1. Random Access.*

This feature enables you to select and revise any step of a program with the following simple procedure.

1. If program is running (RUN LED on), press STOP button (3).
2. Press SELECT STEP button (8). Pointer LED will light.
3. Input desired step number on keypad (4). The selected step number will appear in the STEP display (10).
4. Press desired parameter SELECT button.

*3.6.1.2. Sequential Access.*

In sequential access, you simply use the STEP REV (7) or STEP FWD (9) button to move the program pointer to the step you want to change.
Pressing the button momentarily moves the pointer one parameter; holding the button down causes the pointer to slew rapidly thru parameters and steps.

### 3.6.2. Make the Change.

To make the change, proceed as follows:

1. Turn the MODE SELECT keyswitch (1) to PROGRAM position (1B).
2. Push the applicable SELECT button (random access), or use STEP REV (7) or STEP FWD (9) button, to select the parameter you want to change.
3. The present value will appear on the display for the selected parameter. Input the desired change on the keypad (4), then press ENTER button (6). The change is now recorded.
4. Repeat Steps 2 and 3 as required for each change you wish to make.

## 3.7. RUN THE PROGRAM

This section describes operations where all commands originate from a program in the programmer's memory evoked from the front panel by the operator. No computer or external control signals are used, and the MODE SELECT keyswitch (1) is in the OPERATE position (1A).

### 3.7.1. Access a Step in Operate Mode.

You can access step(s) from the operate mode by either random access or sequential selection. If the programmer is in OPERATE-RUN mode, press STOP button (3) to freeze operation before accessing the new step. The RUN LED will go off.

#### *3.7.1.1. Random Access.*

This feature enables you to select and start at any step of a program with the following simple procedure.

1. Press SELECT STEP button (8). Pointer will light.
2. Input desired step number on keypad (4). The selected step number will appear in the STEP display (10).
3. Push the RUN button (2) to start operation.

*IMPORTANT*

Reviewing a running program using random access will reset the time/cycles clock. Returning to any step and restarting the program will invoke the entire time programmed for that step (program will not resume at the point where it was stopped).

#### *3.7.1.2. Sequential Access.*

With the programmer in STOP mode, sequential access allows you to use the STEP REV (7) or STEP FWD (9) button to move the program pointer to the step you want to review. Pressing these buttons momentarily will cause the pointer to advance one parameter; holding them down will cause the pointer to move rapidly through the parameters and steps.

*IMPORTANT*

When using sequential access, pressing the RUN button will return the program to the step where it was stopped, and the program will resume with the time remaining for that step. DO NOT use the SELECT STEP switch if you're only reviewing parameters; if you do, you can't return to the original step.

### 3.7.2. Start Operation.

To start programmed controller operation, proceed as follows.

1. Set MODE SELECT keyswitch (1) to OPERATE position.
2. Select starting step using either of the methods explained in Para. 3.7.1.

3. Press RUN button. Colon (:) in TIME/CYCLES display (19) will start flashing. The programmer will execute programmed controller operations without any further operator intervention. If sequential access was used in 3.7.1, execution will resume from the point where the program was stopped. If random access was used, execution will begin at the currently displayed step.
4. If an alarm sounds, respond as specified in Para. 3.7.4 or 3.7.5.

### 3.7.3. Monitor Displays.

*3.7.3.1. Active Displays.*

With the keyswitch in OPERATE position and the RUN button pressed (RUN LED is on), the programmer displays current STEP number, realtime current setpoints, actual time remaining, EVENTS status (programmed events have a lighted LED), and the current process temperature (or value) for each operating controller channel. The INCREASE or DECREASE LED on the controller front panel will also light whenever heating or cooling (respectively) is in process.

*3.7.3.2. Reviewing Programmed Values for Current Step.*

*Setpoint* -- To view programmed setpoints in the corresponding CHANNEL display (13 or 16), press and hold down the corresponding SELECT SETPNT button (11 or 14).

*Alarm Limits* -- To view programmed alarm limits in the corresponding CHANNEL display (13 or 16), press and hold down the corresponding SELECT ALARM LIMITS button (12 or 15).

*Duration* -- To view programmed step duration (total time) on the TIME/ CYCLES display (19), press and hold down the SELECT TIME button (17).

*Cycles or Loops* -- To view cycles remaining on the TIME/CYCLES display (19), press and hold down the END PRGM/CYCLES button (18) during any step that is not in a loop. To view the number of loops remaining, press and hold down the END PRGM/CYCLES button during any step that is in a loop. If you hold down the button while the programmer is in RUN, the display will show the number of cycles (or loops) left until completion. If you hold down the button while the programmer is in STOP, the display will show the number of cycles (or loops) originally programmed. In either case, if the number of *cycles* is displayed, the CYCLES LED will light; if the number of *loops* is displayed, the LED will not light.

### 3.7.4. Respond to Deviation Alarm.

When the process variable is outside the range allowed by the selected alarm limits, the following actions occur.

1. The SELECT ALARM LIMITS LED flashes.
2. The applicable channel's alarm output to TB3-10 or -11 goes low.
3. If the program was running and the CONTINUE TO RUN DURING DEV switch is off, the colon (:) in the TIME/CYCLES display stops flashing to indicate that the program has stopped.

*3.7.4.1. Run During Deviation.*

If the CONTINUE TO RUN DURING DEVIATION/STOP switch (Z33-1) is set to ON, the programmer will continue running even though alarm limits have been exceeded. The alarm outputs will either latch on (continue) or automatically reset (turn off), depending on ALARM AUTO RESET switch position (next paragraph).

*3.7.4.2. Reset.*

If the DEVIATION ALARM AUTO RESET switch (Z33-2) is set ON to enable automatic reset, the programmer will automatically resume operation and the alarm outputs will be automatically reset as soon as the out-of-limits condition has ended. If the switch is set OFF to disable the automatic reset, alarm actions continue until parameters return within limits and the RUN button is actuated (either manually or by remote computer).

### 3.7.5. Respond to Probe Open Indication.

If a probe is defective, disconnected, or has an open junction, the following actions occur.

1. The CODE "P. OP." appears in the applicable controller's PROCESS display and the programmer's SELECT SETPNT LED flashes.
2. Heat and Cool outputs automatically turn off.
3. The alarm output is activated.
4. Program execution stops.

Press the STOP button, correct the error condition, and press RUN to resume program operation.

### 3.7.6. Respond to Invalid Setpoint Indication.

If the realtime setpoint goes outside the operating range for the connected controllers, the following actions occur.

1. The SELECT SETPNT LED flashes.
2. A series of dashes (---.-) appears in the controller's PROCESS display.
3. The Heat and Cool outputs are disabled.
4. The alarm output is activated.
5. Program execution stops.

Stop the program and revise the setpoint to one within controller limits, then restart.

## 3.8. STOP THE PROGRAM FOR REVIEW.

In programmed operation, the program will stop automatically when completed. To stop the program during operation for review, simply press the STOP button (3). RUN button LED will go out and the colon on the TIME/CYCLES display (19) will stop flashing to show that program was interrupted. The clock stops; the program freezes at present conditions and holds there until the RUN button (2) is pressed to resume programmer operation. Program parameters can then be reviewed as follows.

1. Use STEP REV or STEP FWD buttons (7 or 9) to move program pointer to desired parameter as indicated by lighted LED above corresponding SELECT button.
2. Value will appear in the display window above the selected parameter: SETPNT, Channel 1 or 2 (11 or 14); ALARM LIMITS, Channel 1 or 2 (12 or 15); and TIME (17). Programmed events are always displayed (their corresponding EVENTS LEDs are on).
3. To review settings at other steps, use random or sequential access (Para. 3.7.1) to select the desired step.

*NOTE*

> Use sequential access if you want to resume the program from the step and time at which it was stopped. Using random access or changing the keyswitch position resets the time/cycles clock.

## 3.9. RESTART PROGRAM.

### 3.9.1. Soft Start.

This feature can be used to abort and restart a program without thermal stress to the device under test. For the soft start to work, the step selected for restart must be one with a nonzero time (time other than 0:00). (The soft start feature calculates a ramp from the current actual process temperature to the programmed temperature based on time allotted for a step.) When the programmer is started from any program step with a nonzero time, the soft start is invoked as follows.

1. The programmer reads the actual *process* temperature and uses that temperature as the starting setpoint for that step.
2. The programmer then takes the process temperature to the programmed value at a linear rate over the time programmed for the ramp.
3. If the program is stopped for any reason, the timer stops and the present setpoint is latched and maintained.

### 3.9.2. Restarting at Ramp or Soak Step (XX:XX STEP TIME).

The setpoint starts at the actual process value and is linearly incremented or decremented to arrive at the final programmed setpoint value at the end of the programmed time.

### 3.9.3. Restarting Without Ramp or Soak (00:00 STEP TIME).

The programmer will supply the controller(s) with the setpoint stored for the restart step. The system will go to the programmed temperature at the maximum possible rate.

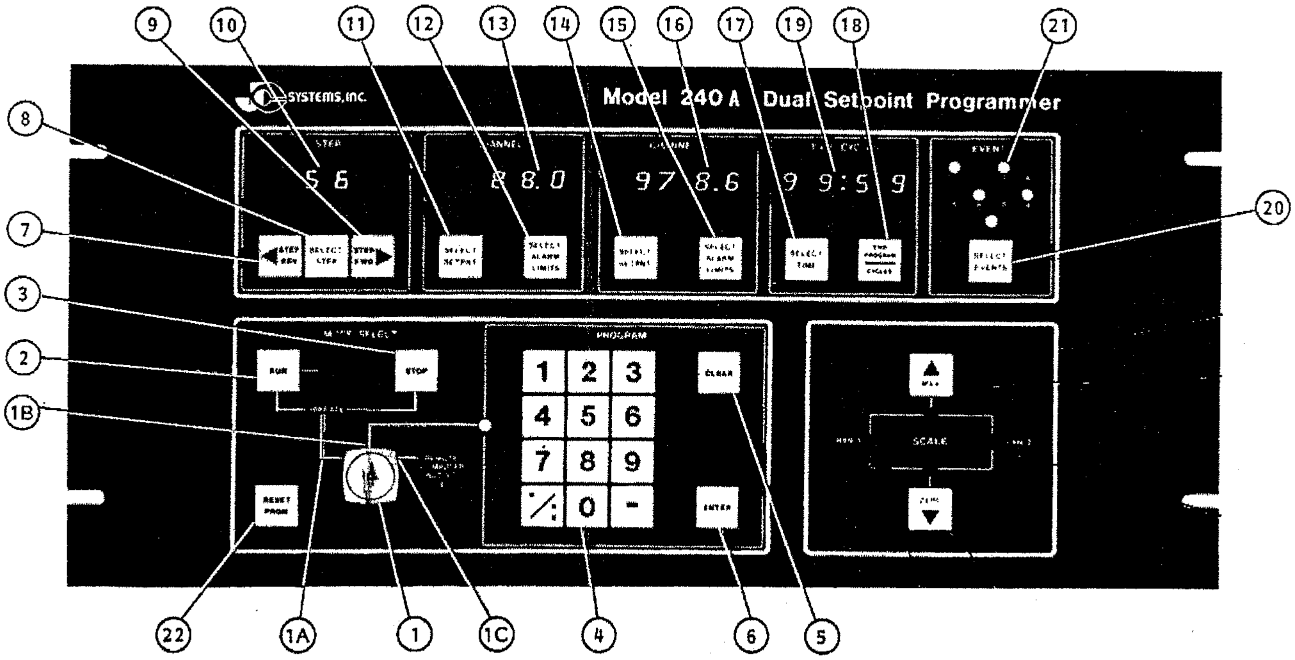
### 3.9.4. Select Restarting Step.

#### *3.9.4.1. Restart at First Step of Present Program.*

To restart at the current program's first step (start the program over), press STOP (3) to stop program operation, then press RESET PRGM button (22) to reset to the first step of the current program, and finally press the RUN button (2) to restart.

#### *3.9.4.2. Restart at Any Other Step.*

To restart at any other step of the same program *OR* a different program, use random access to select the starting step as described in Para. 3.7.1, then press the RUN button (2).



*NOTE:* This duplicate of Figure 3-3 is provided for convenient reference. Fold out this page and leave it open while reading procedures that reference Figure 3-3 callouts.

*Figure 3-3: Front Panel Buttons, Displays and Indicators.*

# Chapter 4. Using the PROMSAVE™ Accessory.

## 4.1. INFORMATION PROVIDED.

This chapter describes how to connect and use the JC Systems Model 285 PromSave™ (Figure 4-1). This device is an optional accessory that permits fast, easy storage and retrieval of programs developed on the JCS Model 240A, 510 or 520. Such programs can be stored on the PromSave™'s cartridge and retrieved for reuse at any time.

## 4.2. HOW THE PROMSAVE™ WORKS.

The PromSave™ uses an EEPROM cartridge that stores programs developed for the user's JC Systems programmer applications. Programs in a RS-232C serial-capable JC Systems programmer such as the Model 240A, 510, or 520 can be readily saved to the Model 285's cartridge EEPROM, stored indefinitely, then loaded back into the programmer when next required. The 285 does an auto-compare and verification between the programmer memory and the cartridge with every memory transfer.

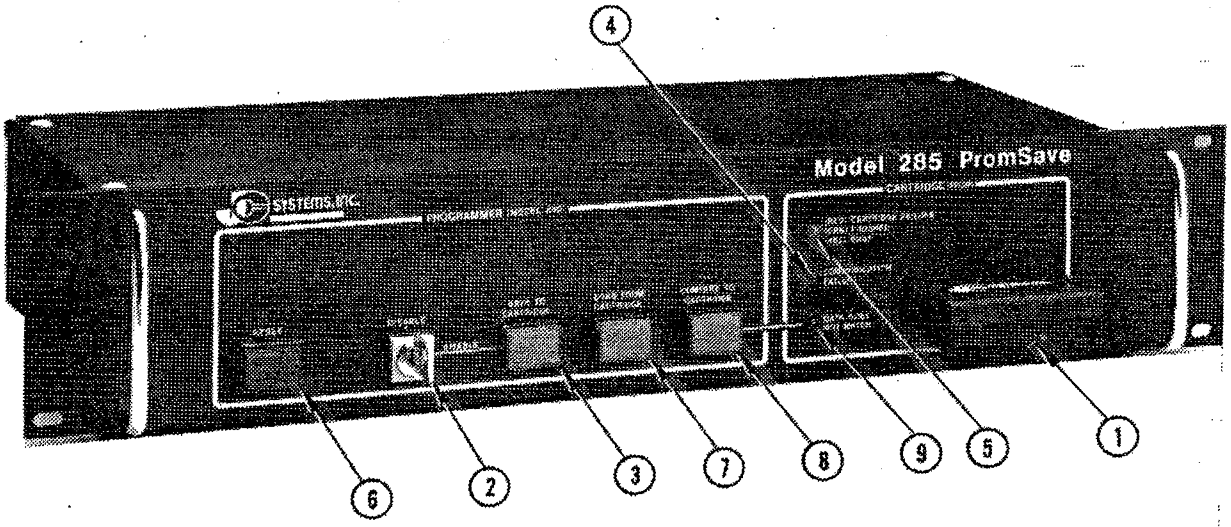


*Figure 4-1: Model 285 PromSave™ Controls and Indicators.*

## 4.3. CONNECT PROMSAVE™ POWER.

The Model 285 requires 115 Vac electrical power at 50-60 Hz.

## 4.4. ESTABLISH COMMUNICATIONS INTERFACE.

### 4.4.1. Connect PromSave™ to Programmer.

Disconnect RS-232C interface from the programmer to any other system at programmer rear panel connector J4. Use the cable supplied with the PromSave™ to connect the programmer to the PromSave™ RS-232C interface on the rear panel at J1, a 25-pin Type D subminiature connector.

### 4.4.2. Configure Programmer for Communication.

*4.4.2.1. RS232C Configuration.*

The programmer standard configuration is RS-232C compatible, so it is ready for use with the PromSave™.

*4.4.2.2. RS422A Configuration.*

If the optional RS-422A/485 Programmer Interface Board A1566 is installed, set switch S3 on that board to the RS-232C position.

*4.4.2.3. IEEE-488 Configuration.*

If the optional IEEE-488 Interface is installed, be sure the Serial Mode Select Jumper is installed on the 2-pin terminal strip on the rear panel.

### 4.4.3. Set programmer to Remote.

Turn MODE SELECT keyswitch on programmer front panel to REMOTE COMPUTER ACTIVE position.

## 4.5. TURN ON PROMSAVE™ POWER.

Press rocker switch on the PromSave™ rear panel to turn on power.

## 4.6. SAVE PROGRAM TO CARTRIDGE.

To save a program stored in memory to the PromSave™ EEPROM cartridge, proceed as follows.

1. Insert EEPROM cartridge (Figure 4-1 Item 1) into its receptacle on PromSave™ front panel.
2. Turn PromSave™ keyswitch (2) to ENABLE position.
3. Press SAVE TO CARTRIDGE button (3). Program transfer will begin. During memory transfer, the PromSave™ will automatically perform a comparison and verification between the programmer memory and the cartridge. One of the three LEDs on the PromSave™ front panel will light as appropriate to indicate cartridge or transfer status as follows.

   a. Top LED (5) will be:
      *Yellow* if PromSave™ is BUSY (transfer in progress).
      *Red* if the cartridge is defective. (Replace cartridge and try again.)
      *Green* when the transfer is successfully completed.

   b. Center LED (4), which is red, will light if there is a communications failure between the programmer and the PromSave™. Check programmer front panel keyswitch (which should be in the REMOTE COMPUTER ACTIVE position), communication interface setup (see Para. 4.4), and RS-232C interface cable connections.

   c. Bottom LED (9), which is also red, will light if the programmer memory and the PromSave™ cartridge memory do not contain the same information.

4. If a red LED lights during transfer, correct the indicated problem and press the RESET button (6) to try again.
5. After transfer has been successfully completed, turn the PromSave™ keyswitch (2) to DISABLE position.

## 4.7. LOAD PROGRAM FROM CARTRIDGE.

To load a program from the PromSave™ EEPROM cartridge to the programmer memory, proceed as follows.

1. Insert EEPROM cartridge and turn PromSave™ keyswitch (2) to DISABLE position.
3. Press LOAD FROM CARTRIDGE button (7). Program transfer will take place in the same manner and with the same indications as described for the SAVE TO CARTRIDGE procedure.

## 4.8. COMPARE CARTRIDGE AND PROGRAMMER MEMORY CONTENTS.

To compare the contents of the two memory storage areas (EEPROM cartridge and programmer memory), set keyswitch to DISABLE and press COMPARE TO CARTRIDGE button (8). If the two are not exactly alike, the DATA DOES NOT MATCH red LED (9) will light.

*NOTE*

> Data comparison occurs automatically when data is transferred from one storage area to another.

## 4.9. INTERRUPT PROMSAVE™ (RESET).

To abort memory transfer, press the RESET button (6). Transfer will stop and the PromSave™ will be returned to its initial state.

To clear a red LED indication, press the RESET button (6).

# Chapter 5. Remote Programming.

## 5.1. INFORMATION PROVIDED.

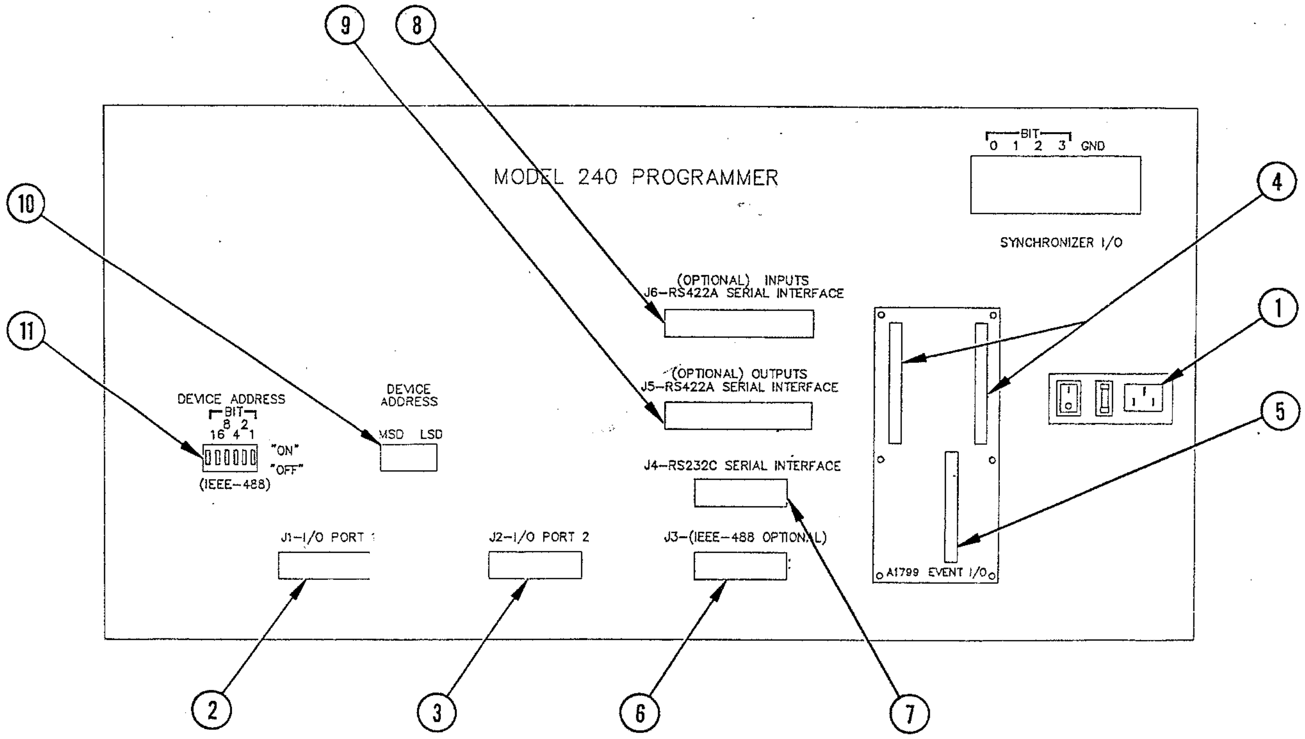
This chapter describes how to interface a remote computer to the programmer and then enter, run, and monitor programs. The programmer understands and responds to commands from a remote computer in the same manner as though you were entering the command directly, using the programmer front panel. (You may want to review Chapter 3 before using the information in this chapter for the first time.) JC Systems has developed PC240 Communications Utility, a BASIC MS-DOS program with a wide range of useful features; if you didn't receive one with your programmer, call Customer Service to request your copy. If you prefer, you may develop your own software using the command set provided in this chapter to communicate with software resident in the programmer.

Note that the RS-232C serial interface capability is standard on any Model 240A, while either the RS-422A/485 interface or the IEEE-488 (GPIB) interface is available as an option. To quickly determine which interface capabilities your unit has, check the connectors installed on the back panel (see Figure 5-1). Active interfaces have a connector mounted in the labeled port. A blank panel means your unit doesn't have that interface capability.

## 5.2. OPERATING WITH EXTERNAL CONTROL INPUTS.

With the mode keyswitch set to OPERATE position, the programmer accepts externally generated inputs for program RUN, STOP, and RESET. Connect control inputs as follows.

Connect external RUN, STOP and RESET control signals at appropriate terminals of A1799 Event I/O Board TB1 on the programmer rear panel (Figure 5-1). (See Table 5-1 for terminal assignments.) Note that these are photo-coupled input lines, and the source voltage (TB1-18) must be connected to a +5V power supply. You may either use an external power supply to provide 5V to the photo-isolators, or use the programmer's internal 5V, 50 mA supply (TB1-17).



*NOTE:* For convenient reference, a duplicate of Figure 5-1 is printed on the last page of this chapter so that you can see the figure when that page is folded out.

*Figure 5-1: Rear Panel Terminals and Connectors.*

Para. 5.2 (Cont.)

If the internal power supply is used, jumper TB1-17 (+5V internal supply) to TB1-18 (isolated source input voltage). Connect TB1-14 (RUN), -15 (RESET), or -16 (STOP) to TB1-13 (0Vdc circuit common) momentarily to actuate the RUN, RESET, or STOP functions, respectively. There is no external isolation in this case.

If an external 5V power supply is used, connect its positive (+) output to TB1-18. The RUN, RESET, or STOP function is actuated by momentarily connecting TB1-14, -15, or -16 to the negative (-) output of the external power supply. No connection should be made to TB1-13 or -17 if external isolation is desired.

## TABLE 5-1: EXTERNAL CONTROL SIGNAL CONNECTIONS.

| NO. | FUNCTION |
|---|---|
| 13 | 0 V DC (circuit common) |
| 14 | RUN |
| 15 | RESET to start of program |
| 16 | STOP |
| 17 | + 5 V DC regulated internal logic power supply (50 mA max) |
| 18 | + Isolated source voltage input |

## 5.3. REMOTE COMPUTER INTERFACE VIA RS-232C SERIAL CONNECTION.

### 5.3.1. Configure programmer for RS-232C Communication.

1. If the IEEE-488 Interface Board is installed, install the Serial Mode Select Jumper on the 2-pin terminal board on the rear panel.
2. Connect a suitable cable from the remote computer to programmer RS-232C port J4, a 25-pin D subminiature connector. The fixed baud rate for this serial interface is 2400. Parity is not required, but odd parity is transmitted. The frame width is 7 bits plus parity plus 1 start bit and 1 stop bit. RS-232C voltage connections are via J4 as follows:

   Transmit - Pin 2     Receive Data - Pin 3     Common - Pin 7

### 5.3.2. Monitoring Programmer Status via RS-232C Interface.

With the programmer MODE SELECT keyswitch in the OPERATE position and either the RUN or STOP mode selected, you can use the remote computer's RS-232C serial interface to monitor the current status of a program. (If you want to take complete control of the programmer from the remote computer, refer to Para. 5.3.3.)

To monitor current status, send an ASCII string command code selected from those shown in Table 5-2. For example, to read the current setpoint on Channel 1, send the ASCII string SP1. To then read the time remaining in the current step, send the ASCII string TIM. The programmer will transmit the requested data immediately upon receipt of the command code. The computer must read (input) the data as a string variable.

*5.3.2.1. Parameter Response.*

A parameter response shows the current setting in numeric form; for example, 75.0 for current setpoint. When a parameter has been cleared (for example, a setpoint for Channel 2 when that controller is not in use), the display will read "---.-". Parameter responses include the setpoints for Channels 1 and 2, as well as alarm limits.

*5.3.2.2. Mode Response.*

A mode response shows a 3-letter abbreviation for the mode; for example, RUN for running, STP for stopped, or EOP for end of program.

#### *5.3.2.3. DIP Switch Position Response.*

A DIP switch response shows the settings of switch positions (bits), reading from left to right as 1 through 8. If the bit is not selected, a 0 is shown; if it is selected, the bit position is shown. For example, a response of 02000000 shows that bit 2 is on and all others are off.

#### *5.3.2.4. Alarm Condition Response.*

The alarm condition response is an eight-character ASCII string as listed below.

L1 = Channel 1 limit exceeded    L2 = Channel 2 limit exceeded
S1 = Controller 1 failure    S2 = Controller 2 failure
"--" replaces code for inactive alarms

For example, if all alarms are active, the response would be "S1L1S2L2"

### 5.3.3. Full Remote Computer Control via RS-232C Interface.

#### *5.3.3.1. How It Works.*

With the programmer SELECT MODE keyswitch in the OPERATE position, the remote computer can monitor the programmer. To transfer full control of the programmer, set the SELECT MODE keyswitch to the REMOTE COMPUTER ACTIVE position. In this mode, the programmer keypad is inactive; all operations are controlled by commands from the interfaced computer. However, all displays and values are shown in the appropriate position on the programmer front panel when the programmer is stopped.

In addition to the mode commands for any switch position listed in Table 5-2, the operational commands listed in Table 5-3 are accepted by the programmer only in REMOTE COMPUTER ACTIVE mode. All commands in Table 5-3 except "!" require the keyswitch to be in the REMOTE COMPUTER ACTIVE position. "!" can be used to enter a step number under any circumstances. Other "!" entries require that programmer be stopped and in REMOTE COMPUTER ACTIVE.

## TABLE 5-2: COMMAND SET FOR OPERATE OR REMOTE COMPUTER ACTIVE KEYSWITCH POSITION

| COMMAND | FUNCTION |
|---|---|
| SP1 | Read current setpoint #1 |
| SP2 | Read current setpoint #2 |
| PR1 | Read current process variable #1 |
| PR2 | Read current process variable #2 |
| TIM | Read time to go in current step |
| STN | Read current step number |
| CTG | Read number of cycles to go in current program |
| LTG | Read number of loops to go in current nested loop |
| DIP | Read internal programmer and configuration DIP switch setting (Z33) |
| SW1 | Read internal DIP switch 1 (same as DIP) |
| SW2 | Read internal DIP switch 2 (S1 on A1852/1550) |
| LCK | Read position of keyswitch |
| ALM | Read alarm status |
| SS1 | Select Setpoint 1 |
| SS2 | Select Setpoint 2 |
| SL1 | Select Limit 1 |
| SL2 | Select Limit 2 |
| STM | Select Time |
| SE1 | Select stored events |
| EV1 | Select event outputs |
| SST | Select Step Mode |
| SCY | Select Cycles/End of Program |
| SLP | Select Loops |
| RQN | Service request enable (IEEE-488 only) |
| RQD | Service request disable (IEEE-488 only) |
| MOD | Read mode (Run, Stop, or End of Program) |
| ? | Used during programming to read program parameters and to advance program pointer one step at a time |
| ! | Enter step number and advance pointer |
| ULD | Upload - send entire programmer memory contents to the computer (programmer must be in STOP mode) |
| PRR | Reset to first step in program being reviewed (Same as pressing RESET button) |

## TABLE 5-3: COMMAND SET FOR REMOTE COMPUTER ACTIVE KEYSWITCH POSITION.

| COMMAND | FUNCTION |
|---|---|
| RUN | Start operation (Same as pressing RUN button) |
| STP | Stop operation (Same as pressing STOP button) |
| PRR | Reset to first step in present program (Same as pressing RESET button) |
| ! | Enter data (same as pressing ENTER button) |
| CLR | Clear data (same as pressing CLEAR button) |
| CTL | Read controller modes |
| MN1 | Place controller 1 into manual mode |
| MN2 | Place controller 2 into manual mode |
| PG1 | Place controller 1 into program mode |
| PG2 | Place controller 2 into program mode |
| DLD | Download command - accepts data in the same format that was produced as a result of the ULD (upload) command and loads programmer memory with all information previously uploaded to the computer. Accepts 2050 characters; spaces ( ) and carriage returns <↵> are not counted in total. |

*5.3.3.2. Select a Step.*

There are three ways to access a desired step.

1. If you are currently in the Stop mode n any step of a program, you can return to the first step of that program by sending the reset command, PRR.
2. If you know the step number you want, use the random access feature to select that step as follows:
   a. Send the Select Step command SST.
   b. Send the desired step number, followed by an exclamation mark (!). This is the same as pressing the ENTER button in direct manual operation.

*5.3.3.3. Select a Parameter.*

The command set used with full remote computer control of the programmer permits random access to a specific parameter (or the events) within a step. These commands -- SCY, SLP, SS1, SS2, SL1, SL2,STM, and SE1/EV1 -- are listed and defined in Table 5-2. For example, to change the time for a given step, proceed as follows.

1. Access the desired step per 5.3.3.2.
2. Send the Select Time command string STM.
3. Send the new time as explained in 5.3.3.4.

*5.3.3.4. Enter a Parameter.*

To enter a new or changed parameter from a remote computer, note that you must send the new entry, then press the exclamation mark (!). This transmits the data from the remote computer to the programmer. You can send two or more parameters for a step as a single string as follows.

1. Send the first parameter to be programmed or changed followed by the !, which is the remote computer ENTER command.
2. Send the second and subsequent parameters *in the order they appear* (on the front panel display or the program worksheet), separating each by an !.

## 5.4. REMOTE COMPUTER INTERFACE VIA RS-422A/485 SERIAL NETWORK OPTION.

### 5.4.1. Configuring for RS-422A/485 Communication.

A single master computer can communicate with up to 99 programmers that are equipped with the RS-422A/485 option. However, care must be taken to correctly connect the programmer and the master computer. Several switch positions also must be selected based on system configuration. These requirements are detailed in the following paragraphs.

### *5.4.1.1. Voltage Connections.*

Unless your computer is equipped with an RS-422A/485 port, you will need to install an RS-232C/RS-422A/485 level shifter (JC Systems Model A1740 or equivalent) between the RS-232C port of the master computer and the first programmer in the network as shown in Figure 5-2.

RS-422A/485 voltage-level connections from the computer to the first programmer on the party line are via RS-422A/485 Port J6, a 37-pin "D" type miniature connector. (Pin assignments are given in Table 5-4.) Connector J5 on the first programmer then connects to J6 on the second programmer. Subsequent programmers in the network connect in the same manner as the first pair.

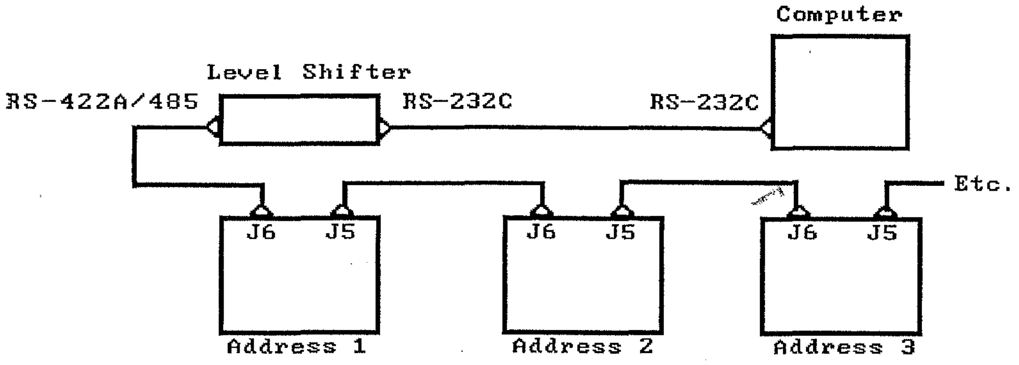


*Figure 5-2: RS-422A/485 Serial Communication Connections.*

## TABLE 5-4: RS-422A/485 PORT PIN ASSIGNMENTS.

| FUNCTION | SIGNAL | COMPUTER TO UNIT #1 J6 PIN NO. | UNIT #1 TO UNIT #2 J5 PIN NO. | UNIT #2 TO UNIT #3 J6 PIN NO. |
|---|---|---|---|---|
| Send Data (to Master Computer) | SD (A)<br>SD (B) | 4<br>22 | 4<br>22 | 4<br>22 |
| Receive Data (from Master Computer) | RD (A')<br>RD (B') | 6<br>24 | 6<br>24 | 6<br>24 |
| Signal Ground | SG | 19 | 19 | 19 |

### *5.4.1.2. RS-422A/485 Interface Switch Settings.*

The following switches must be set as specified to ensure correct party-line serial operation. To gain access to internal switches, loosen camloc fasteners securing the top cover and remove the cover.

1. *Address Switch* on programmer rear panel -- this direct-reading rotary switch (Figure 5-1, Item 10) must be set to a different number (address) for each programmer connected to the system. When two or more programmers are used, the usable addresses are 1-99 inclusive.

*NOTE*

> The user will need assigned addresses to communicate command and status information from the master computer to party-line programmers. Therefore, a hard-copy address chart posted near the computer showing the address and physical location of each programmer on the network might prove useful in some applications.

2. *RS-232C/RS-422A/485 Select Switch S3* on photo-isolated A1566 Interface Board (Figure 5-3, Item 1) -- set to RS-422A/485 (up) position for serial communications between a single computer and several programmers on a RS-422A/485 network.

*NOTE*

> An RS-232C/RS-422A/485 Level Shifter (JC Systems Model A1740 or equivalent) must be used to convert RS-232C output from the computer for RS-422A/485 network use.

3. *Baud Rate Dip Switch S1* on A1566 (Item 2) -- Select a baud rate of 300, 600, 1200, 2400, 4800, or 9600 by placing that switch in the ON (down) position. (9600 is the maximum baud rate for programmer communications.) All other baud switches must be in the OFF (up) position, and all programers sharing party-line serial communications must be set to the same rate.

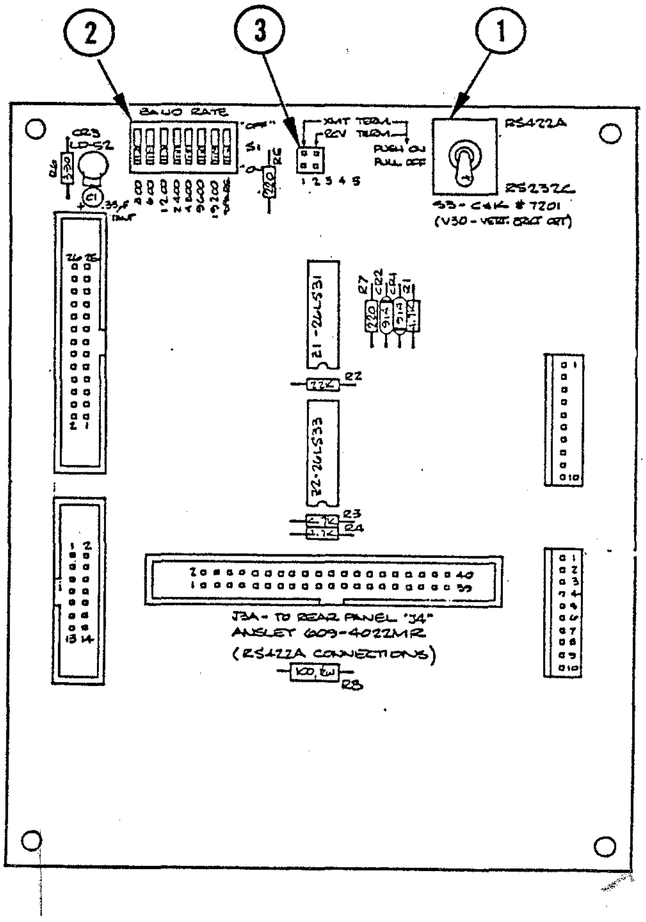


*Figure 5-3: RS422A Programmer Interface Board A1566.*

### 5.4.2. Monitoring Status via RS-422A/485 Interface.

With the MODE SELECT keyswitch in the OPERATE position, you can use a remote computer's serial interface to monitor the current status of a program. To monitor current status, send an ASCII string command code selected from those shown in Table 5-2. For example, to read the current setpoint on Channel 1, send the ASCII string SP1. To then read the time remaining in the current step, send the ASCII string TIM. The programmer will transmit the requested data immediately upon receipt of the command code. The computer must read (input) the data into a string variable.

### 5.4.3. Full Remote Computer Control via RS-422A/485 Interface.

With the SELECT MODE keyswitch in the OPERATE position, the remote computer is used to read values already programmed. To transfer full control of the programmer, set the SELECT MODE keyswitch in the REMOTE COMPUTER ACTIVE position. In this mode, the front panel controls are inactive; all operations are controlled by commands from the interfaced computer.

### 5.4.4. Format for Entering RS-422A/485 Computer Commands.

You must send the address of the specific programmer you want to query before you send the command code string. To enter the address, send the disconnect command, a number symbol (#), followed by the address number assigned to the programmer (1-99). **There must be a blank space between the address number and the command (see Figure 5-4).**

The disconnect command (#) disconnects all addresses, clearing the network. The programmer address number connects the computer to the selected address *only*, and the computer remains connected to that address until another disconnect command is received. Therefore, you only need to transmit the address once, immediately after the disconnect command. The computer will continue to address the same programmer from that time until the next disconnect command is sent.

```
EXAMPLE:  #3 SP1
          ||| |
          ||| |__ Command (from Table 5-2 or 5-3)
          |||____ Blank space
          ||_____ Address number
          |______ Number symbol (disconnect command)
```

*Figure 5-4: Format for Remote Computer Commands (more than one programmer on line).*

## 5.5. REMOTE COMPUTER INTERFACE VIA IEEE-488 (GPIB) OPTION.

This optional interface provides a remote IEEE-488 controller or computer with the same command and communications capabilities as the RS-232C interface described in Para. 5.3. In addition, the IEEE-488 option includes the capability to generate service requests upon user-selected events.

### 5.5.1. Configure for IEEE-488 Interface.

If the IEEE-488 computer interface option is installed, connect IEEE-488 system at Connector J3 on the rear panel (Figure 5-1 Item 7), make sure the Serial Mode Select Jumper is not installed on the two-pin terminal board on the rear panel, and set service request function and address switches as specified below.

### *5.5.1.1. Set Service Request DIP Switch.*

The IEEE-488 option includes a service request feature which, when enabled, causes the programmer to generate a service request when a deviation alarm condition begins and/or ends. The user can also set switches on the SRQ board, which is mounted on the inside of the rear panel, to generate a service request at selected events.

Service request DIP switch S1 positions on the SRQ and device address interface board A1848 (Figure 5-5) are marked YES/NO. You can select what conditions will (or will not) cause a service request by the switch positions. Selecting the YES position causes a service request to be generated when the specified condition exists; the NO position prevents the service request. Selecting the YES position for both the Event ON and Event OFF will generate a service request both when the event turns ON and when it turns OFF.

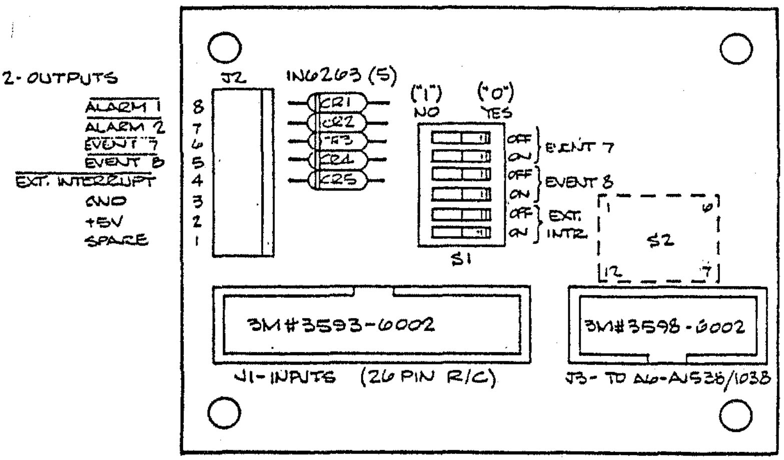


*Figure 5-5: IEEE-488 SRQ and Device Address Board A1848.*

Para. 5.5.1.1 (Cont.)

Service request DIP switch positions and corresponding events are identified below.

1. Positions 1 and 2 control service requests when Event 7 turns OFF and ON, respectively.
2. Positions 3 and 4 control service requests when Event 8 turns OFF and ON, respectively.

*NOTE*

> Event 8 and the External Interrupt (bit positions 5 and 6 on the DIP switch) share common access to the IEEE-488 interface, so do not select both at the same time.

3. Positions 5 and 6 allow the External Interrupt Input access to the IEEE-488 interface. Connecting TB3 Pin 19 (remote interrupt) to Pin 13 (ground) with an isolated set of contacts generates a service request if SRQ board switch S1 positions 5 and 6 are both set to YES.

### *5.5.1.2. Read Service Request (Serial Poll) Data.*

All programmers on the buss are polled by the computer to identify the one that generated the service request and to determine what condition(s) caused the service request to be generated.

The serial poll response from the programmers is an 8-bit word in ASCII characters that gives the computer this information. The list below explains the significance of each bit setting. Note that Bits 0 and 1, for example, work together to show not only if a Channel 1 alarm is on or off (bit 0), but also if the alarm status has changed (bit 1) since the last service request. The other bits provide similar information. After a serial poll, all bits are automatically reset.

```
Bit 0 set -- Channel 1 Alarm is ON
Bit 1 set -- Channel 1 Alarm status change
Bit 2 set -- Channel 2 Alarm is ON
Bit 3 set -- Channel 2 Alarm status change
Bit 4 set -- Event 7 ON
Bit 5 set -- Event 8 ON
Bit 6 set -- When any condition causes a Service Request
Bit 7 set -- Programmer failure caused a Service Request
```

*5.5.1.3. Set Device Address.*

The IEEE-488 device address switch (S2 on the A1848 PCB, which is accessible on the programmer rear panel -- see Figure 5-1, Item 11) must be set to a different number (address) for each programmer connected to the network. The highest possible address is 31 (all bits set to ON); however, address 31 is normally reserved for UNTALK and UNLISTEN commands.

### 5.5.2. Remote Computer Operations via the IEEE-488 Interface.

*5.5.2.1. Commands and Format.*

Commands and command format for the IEEE-488 interface are the same as those previously described for the RS-422A/485 interface (see Para. 5.4), except as follows:

1. The service request functions have been added as described below.
2. "#" (address) is not part of the transmitted command string as it is with the RS-422A/485.

*5.5.2.2. Service Request Enable/Disable.*

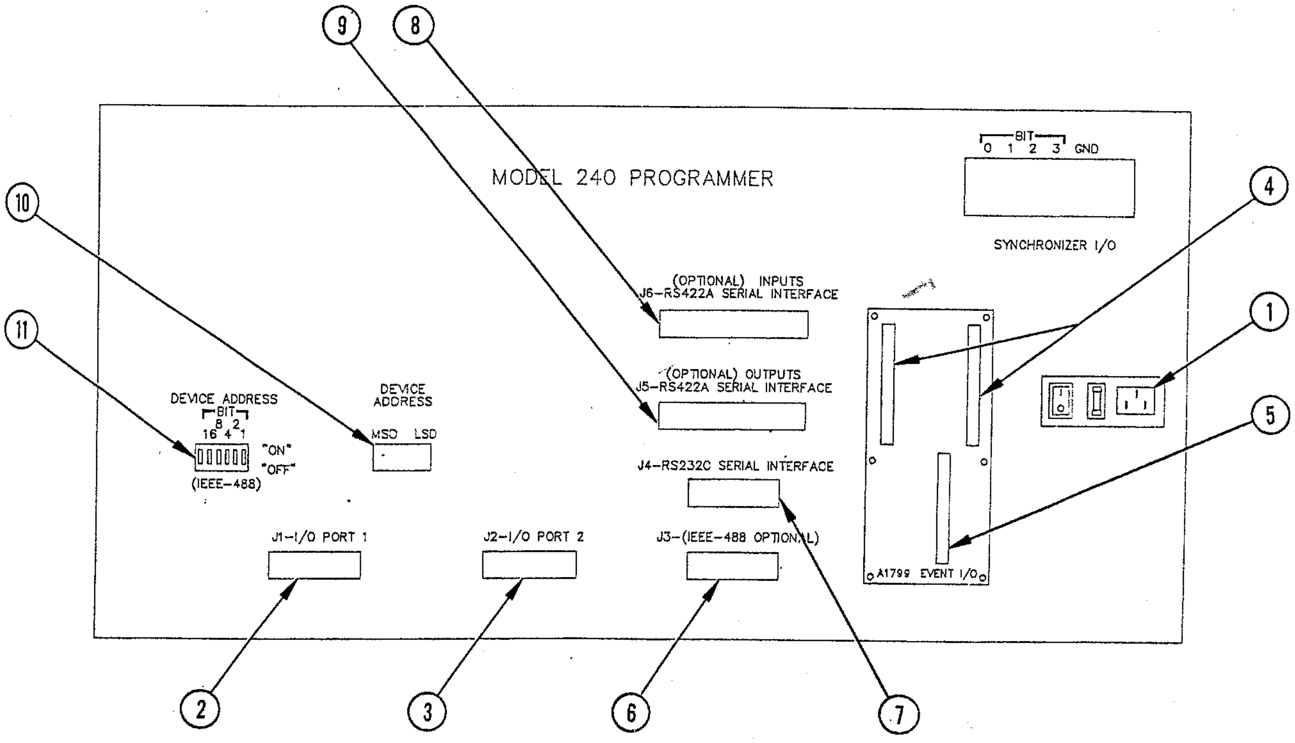
To enable the service request function, an RQN command must be transmitted by the computer. An RQD command disables the service request.

| | |
|---|---|
| *RQN | Service Request Enable |
| *RQD | Service Request Disable |

* Active only with IEEE-488 option. When installed as described in Para. 5.5.1, the service request capability can generate an interrupt whenever the alarm status for either channel changes. These commands turn that capability on and off.

*NOTE*

> There is no memory of service request conditions that occur while the service request is disabled. Therefore, when the service request is enabled, a service request will not be generated for a condition that occurred while the service request was disabled.



*NOTE:* This duplicate of Figure 5-1 is provided for convenient reference. Fold out this page and leave it open while reading procedures that reference Figure 5-1 callouts.

*Figure 5-1: Rear Panel Terminals and Connectors.*

# Chapter 6. Maintenance and Calibration.

## 6.1. PREVENTIVE MAINTENANCE.

### 6.1.1. Cleaning.

Regularly dust outside surfaces and keeping the programmer's interior free of dust and debris, especially pieces of wire that could cause shorts.

### 6.1.2. Other Preventive Maintenance.

1. Perform the calibration procedures described in this chapter at least once a year.
2. Replace the lithium battery every five years.

## 6.2. PERFORM CORRECTIVE MAINTENANCE.

The programmer is designed to be trouble-free, offering reliable service without extensive maintenance. The only corrective maintenance recommended consists of troubleshooting to the board level using procedures provided in this chapter, followed by removal and replacement of defective components if required.

## 6.3. PROGRAMMER TEST PROCEDURES.

### 6.3.1 Power-Up and Threshold Test.

1. Using an ohmmeter, check for shorts between ground and +5V, +9V, and +5V pulsed. All values should be greater than 100 ohms.
2. Connect Variac set for 117Vac RMS.
3. Plug in programmer to Variac and use voltmeter to check voltage readings: +5V should be 4.91 to 5.15V; +9V should be 8.3 to 11V; +5 pulsed should be 2.5 to 3.9V.
4. Connect an oscilloscope set to 2V per division to TP3 on logic control board A1550. Adjust Variac for 85 ±1Vac RMS, then adjust POWER FAIL threshold resistor R24 until pulses just appear on scope. Pulses should disappear when Variac voltage increases and reappear when it decreases.
5. Remove scope from TP3, remove Variac, and restore normal power to programmer.

### 6.3.2 Front Panel Switch and Display Tests.

*NOTE*

Perform these tests after exiting diagnostic mode.

1. Set keyswitch to program mode and press SELECT STEP button (Figure 6-1, Item 8). SELECT STEP LED should light. If STEP display (10) doesn't show the number 1, press 1 on keypad (4), then ENTER button (6); STEP display should now show 1.
2. Press and hold down STEP FWD button (9). STEP display should slew (move rapidly) from 2 to 91. Verify that all step numbers light with uniform brightness.
3. Press and hold down STEP REV button (7). STEP display should slew from 91 back to 1. Press SELECT STEP button (8), then 1 on keypad (4), and finally ENTER button (6); program pointer LED should light above Channel 1 SELECT SETPNT button (11).
4. Press Channel 1 SELECT SETPNT button, then press numbers on keypad (1,2,3, etc.). Numbers on display should shift left from the decimal point. Press decimal point and numbers again; numbers should shift right from decimal point.
5. Press keypad #1 and hold for 1/2 second; key should not auto-repeat.
6. Repeat Steps 4 and 5 with each parameter button and display.
7. Press MAN/PROG button for controller Channels 1 and 2 several times. LED should indicate corresponding changes in mode.

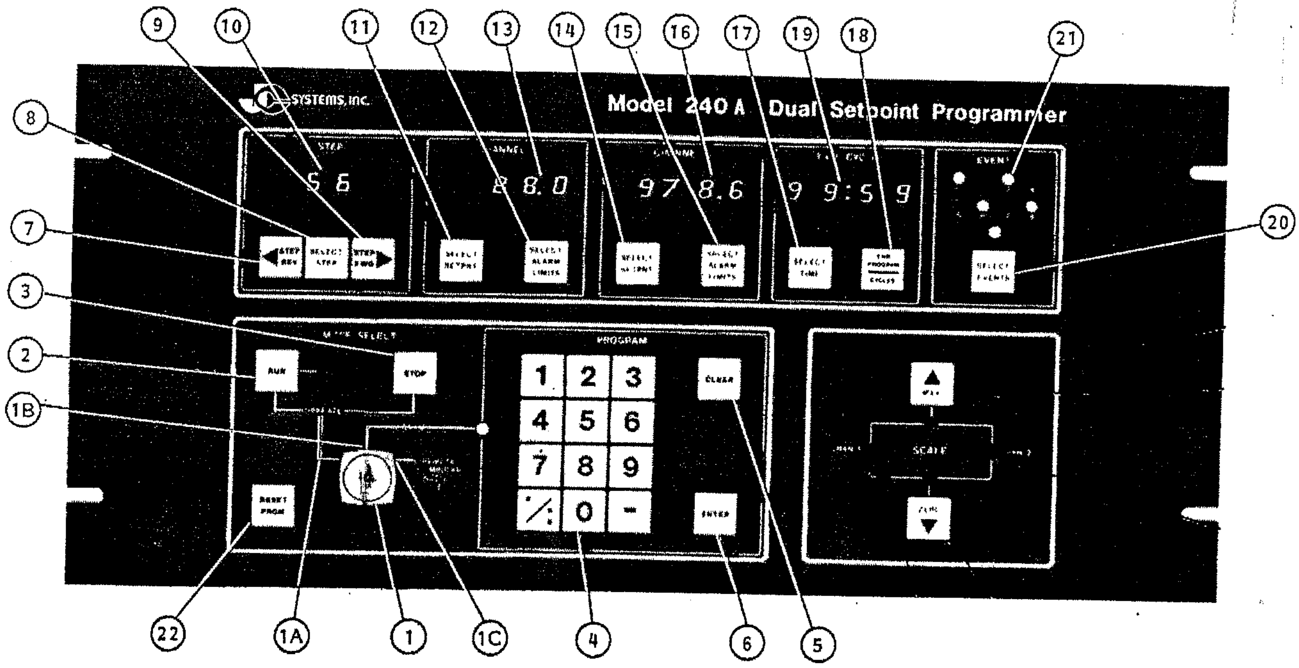


*Figure 6-1: Front Panel Buttons, Displays and Indicators.*

# APPENDIX A. PROGRAM WORKSHEET

JC SYSTEMS INC. PROGRAM NO. __________ DATE: ______________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | |

*NOTES:*

1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

## APPENDIX A. PROGRAM WORKSHEET (Cont.)

JC SYSTEMS INC. PROGRAM NO. __________ DATE: ______________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 31 | | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | | |
| 33 | | | | | | | | | | | | | |
| 34 | | | | | | | | | | | | | |
| 35 | | | | | | | | | | | | | |
| 36 | | | | | | | | | | | | | |
| 37 | | | | | | | | | | | | | |
| 38 | | | | | | | | | | | | | |
| 39 | | | | | | | | | | | | | |
| 40 | | | | | | | | | | | | | |
| 41 | | | | | | | | | | | | | |
| 42 | | | | | | | | | | | | | |
| 43 | | | | | | | | | | | | | |
| 44 | | | | | | | | | | | | | |
| 45 | | | | | | | | | | | | | |
| 46 | | | | | | | | | | | | | |
| 47 | | | | | | | | | | | | | |
| 48 | | | | | | | | | | | | | |
| 49 | | | | | | | | | | | | | |
| 50 | | | | | | | | | | | | | |
| 51 | | | | | | | | | | | | | |
| 52 | | | | | | | | | | | | | |
| 53 | | | | | | | | | | | | | |
| 54 | | | | | | | | | | | | | |
| 55 | | | | | | | | | | | | | |
| 56 | | | | | | | | | | | | | |
| 57 | | | | | | | | | | | | | |
| 58 | | | | | | | | | | | | | |
| 59 | | | | | | | | | | | | | |
| 60 | | | | | | | | | | | | | |

*NOTES:*

1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

## APPENDIX A. PROGRAM WORKSHEET (Cont.)

JC SYSTEMS INC. PROGRAM NO. __________ DATE: ______________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 61 | | | | | | | | | | | | | |
| 62 | | | | | | | | | | | | | |
| 63 | | | | | | | | | | | | | |
| 64 | | | | | | | | | | | | | |
| 65 | | | | | | | | | | | | | |
| 66 | | | | | | | | | | | | | |
| 67 | | | | | | | | | | | | | |
| 68 | | | | | | | | | | | | | |
| 69 | | | | | | | | | | | | | |
| 70 | | | | | | | | | | | | | |
| 71 | | | | | | | | | | | | | |
| 72 | | | | | | | | | | | | | |
| 73 | | | | | | | | | | | | | |
| 74 | | | | | | | | | | | | | |
| 75 | | | | | | | | | | | | | |
| 76 | | | | | | | | | | | | | |
| 77 | | | | | | | | | | | | | |
| 78 | | | | | | | | | | | | | |
| 79 | | | | | | | | | | | | | |
| 80 | | | | | | | | | | | | | |
| 81 | | | | | | | | | | | | | |
| 82 | | | | | | | | | | | | | |
| 83 | | | | | | | | | | | | | |
| 84 | | | | | | | | | | | | | |
| 85 | | | | | | | | | | | | | |
| 86 | | | | | | | | | | | | | |
| 87 | | | | | | | | | | | | | |
| 88 | | | | | | | | | | | | | |
| 89 | | | | | | | | | | | | | |
| 90 | | | | | | | | | | | | | |
| 91 | | | | | | | | | | | | | |

*NOTES:*

1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

# Appendix B. Programming Via an External Computer

## B.1 INTRODUCTION.

The JC Sytems Model 240A, 510 and 520 programmers incorporate a built-in RS232C serial port and are offered with an optional RS422A/485 port or an optional IEEE488 GPIB interface port. This makes it easy to use an external computer with either serial or GPIB capability.

This bulletin provides examples for anyone who wants to write programs and utilize their own computer for program storage, programming and data acquisition.

## B.2 COMMANDS.

Two types of commands are sent to the programmer:

1. *Query* Commands -- elicit an immediate response from the programmer and consequently require that the computer accept the reply from the programmer.

2. *Mode* Commands - Place the programmer in a mode to receive data, or in a mode of operation.

### B.2.1. Query Commands.

All query commands are identified by the phrase "read ---------" in the computer communications command table of the applicable brochure (Tables 5-2 and 5-3 of this manual). For example, the command listing

*SP1 Read current setpoint #1*

means that the programmer will send back the actual current setpoint for channel #1 when it receives the ASCII string *SP1*.

### B.2.2. Mode Commands.

Mode commands do not generate an automatic response from the programmer. An example of a mode command is *SST*. When the programmer receives the ASCII string *SST* it waits for additional information. The additional information in this case is the step number followed by the *!*.

Both the ASCII characters *!* and *?* mean something to the programmer. The *!* is a delimiter that marks the end of a data entry. The *?* is a command to the programmer to return the information that the LED pointer is currently pointing at.

Start the **programmer** by sending the ASCII string *PRINT#1,"RUN"*.

Stop the **programmer** by sending the ASCII string *"STP"*.

Commands for parallel and serial operations are identical except for format changes required by the computer you are using.

## B.3. Sample Programs.

Some examples will help to show how simple it is to program from an external computer.

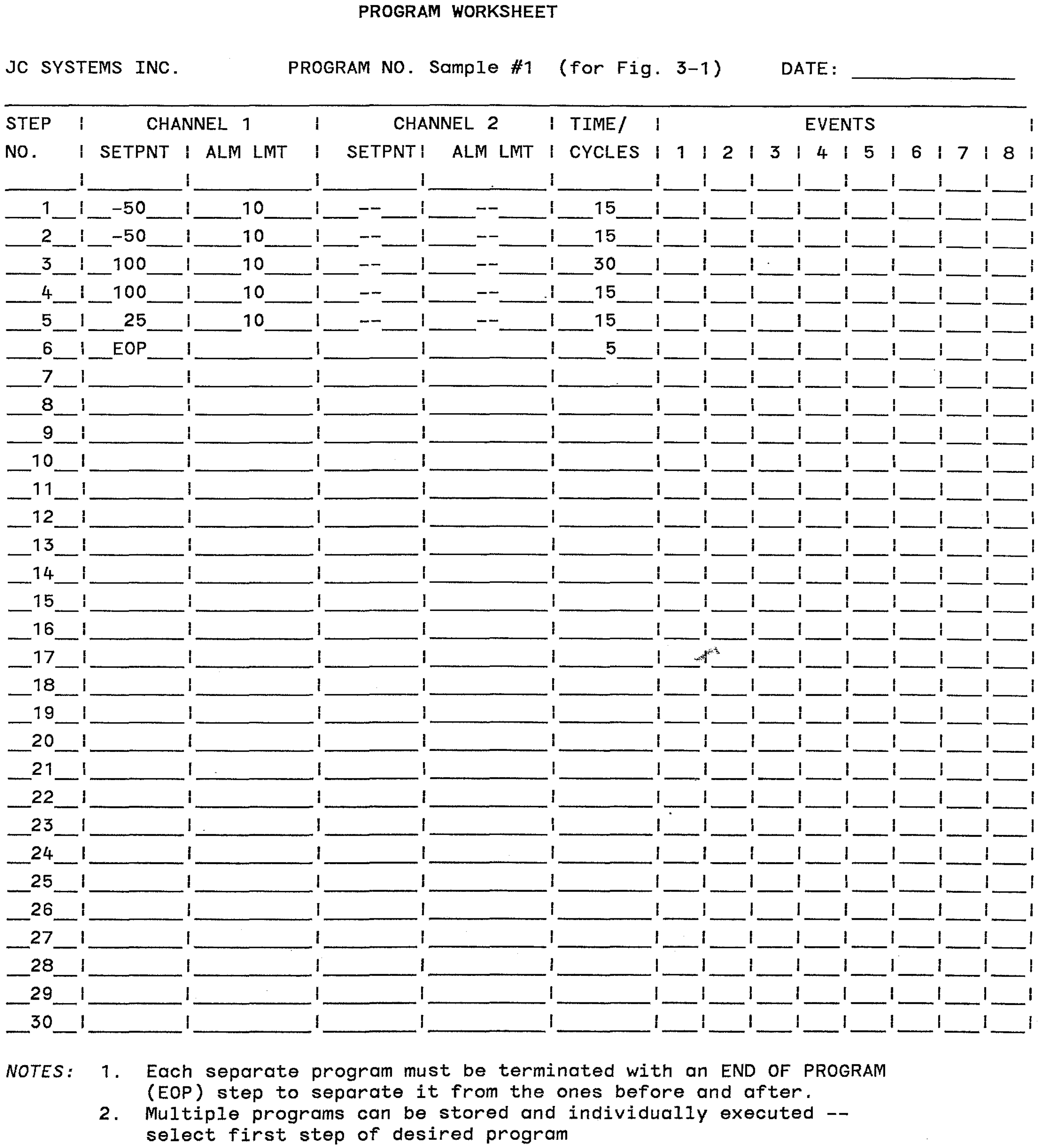
### B.3.1. Sample #1.

Figure B-1 is the program worksheet for the example described in Para. 3.2 of this manual. For simplicity, we show the command structure for a serial interface computer system utilizing MSDOS BASIC and Port #1 with a baud rate of 2400, odd parity, 7 data bits, and 1 stop bit.

Before sending commands to the programmer, the following line must be executed by the BASIC program.

```
OPEN "COM1:2400,O,7,1,RS,DS0" AS 1
```

The above statement opens port #1 as a serial communication port with the correct protocol.



PROGRAM WORKSHEET

JC SYSTEMS INC. PROGRAM NO. Sample #1 (for Fig. 3-1) DATE: ____________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | -50 | 10 | -- | -- | 15 | | | | | | | | |
| 2 | -50 | 10 | -- | -- | 15 | | | | | | | | |
| 3 | 100 | 10 | -- | -- | 30 | | | | | | | | |
| 4 | 100 | 10 | -- | -- | 15 | | | | | | | | |
| 5 | 25 | 10 | -- | -- | 15 | | | | | | | | |
| 6 | EOP | | | | 5 | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | |

*NOTES:*

1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.
2. Multiple programs can be stored and individually executed -- select first step of desired program

*Figure B-1: Program Worksheet with Program Entered.*

Para. B.3.1 (Cont.)

The statement for programming step 1 on a dual-channel unit (the Model 240, for example) is:

*PRINT#1,"SST1!-50!10!CLR!CLR!15!CLR!"*

The first and second *CLR!* commands clear the Channel 2 setpoint and alarm limits because the sample program only uses one of the two available channels. These clear commands would be omitted on a single-channel unit (the Model 510, for example).

Step 2 would be performed by executing the following statement:

*PRINT#1,"SST2!-50!10!CLR!CLR!15!CLR!"*

Step 3 would be performed by executing the statement:

*PRINT#1,"SST3!100!10!CLR!CLR!30!CLR!"*

The complete BASIC program for Example 1 would include the following statements:

```
10 CLEAR:CLOSE
20 OPEN "COM1:2400,O,7,1,RS,DS0" AS 1
30 PRINT#1,"#2 SST1!-50!10!CLR!CLR!15!CLR!"
40 PRINT#1,"SST2!-50!10!CLR!CLR!15!CLR!"
50 PRINT#1,"SST3!100!10!CLR!CLR!30!CLR!"
60 PRINT#1,"SST4!100!10!CLR!CLR!15!CLR!"
70 PRINT#1,"SST5!25!10!CLR!CLR!15!CLR!"
80 PRINT#1,"SST6!SCY5!"
90 END
```

Please note that line 30 specifies the address (in this case Address 2). After the address command has been executed once, it does not have to be repeated unless another programmer is addressed.

If the unit is an RS-232C serial interface (no RS-422A/485 network option), omit the #2 and the space following the 2.

Para. B.3.1 (Cont.)

For a computer like the HP85 with a GPIB interface, the format of a typical line is as follows (assuming that the address of the programmer with the GPIB interface option is 28):

```
10 OUTPUT 728;"SST1!-50!10!CLR!CLR!15!CLR!"
20 OUTPUT 728;"SST2!-50!10!CLR!CLR!15!CLR!"
30 OUTPUT 728;"SST3!100!10!CLR!CLR!30!CLR!"
40 OUTPUT 728;"SST4!100!10!CLR!CLR!15!CLR!"
50 OUTPUT 728;"SST5!25!10!CLR!CLR!15!CLR!"
60 OUTPUT 728;"SST6!SCY5!"
```

### B.3.2. Sample #2.

Figure B-2 is the program worksheet for the example described in Para. 3.4 of this manual.

#### *B.3.2.1. Serial Communications (RS-232C, RS-422A/485).*

```
10 CLEAR:CLOSE
20 OPEN "COM1:2400,O,7,1,RS,DS0" AS 1
30 PRINT#1,"SST10!150!10!CLR!CLR!30!CLR!"
40 PRINT#1,"SST11!SLP1!"
50 PRINT#1,"SST12!150!10!CLR!CLR!10!1!"
60 PRINT#1,"SST13!150!10!CLR!CLR!10!CLR!"
70 PRINT#1,"SST14!SLP50!"
80 PRINT#1,"SST15!25!10!CLR!CLR!30!CLR!"
90 PRINT#1,"SST16!SCY20!"
100 END
```

#### *B.3.2.2. Parallel Communications (IEEE-488/GPIB).*

```
10 OUTPUT 728;"SST10!150!10!CLR!CLR!30!CLR!"
20 OUTPUT 728;"SST11!SLP1!"
30 OUTPUT 728;"SST12!150!10!CLR!CLR!10!1!"
40 OUTPUT 728;"SST13!150!10!CLR!CLR!10!CLR!"
50 OUTPUT 728;"SST14!SLP50!"
60 OUTPUT 728;"SST15!25!10!CLR!CLR!30!CLR!"
70 OUTPUT 728;"SST16!SCY20!"
```

## PROGRAM WORKSHEET

JC SYSTEMS INC. PROGRAM NO. Sample #2 (for Fig. 3-4) DATE: ____________

| STEP | CHANNEL 1 | | CHANNEL 2 | | TIME/ | EVENTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | SETPNT | ALM LMT | SETPNT | ALM LMT | CYCLES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| * 9 | EOP | | | | 1 | | | | | | | | |
| 1Ø | 15Ø | 1Ø | -- | -- | 3Ø | | | | | | | | |
| 11 | LOOP | | | | 1 | | | | | | | | |
| 12 | 15Ø | 1Ø | -- | -- | 1Ø | ON | | | | | | | |
| 13 | 15Ø | 1Ø | -- | -- | 1Ø | | | | | | | | |
| 14 | LOOP | | | | 5Ø | | | | | | | | |
| 15 | 25 | 1Ø | -- | -- | 3Ø | | | | | | | | |
| 16 | EOP | | | | 2Ø | | | | | | | | |
| 17 | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | |
| 2Ø | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | |
| 3Ø | | | | | | | | | | | | | |

*This EOP step used to mark start of program (on Step 1Ø).

@font MX9I;NOTES: 1. Each separate program must be terminated with an END OF PROGRAM (EOP) step to separate it from the ones before and after.

2. Multiple programs can be stored and individually executed -- select first step of desired program

*Figure B-2: Worksheet for Program Sample with Nested Loop.*

# Appendix C. Model A2014 Multi-programmer Synchronizer

## C.1 SUMMARY.

JC System's Model A2014 Multi-Programmer Synchronizer option synchronizes Model 510, 520 and 240A programmers so that all units track on the same time basis. Up to 10 units can be daisy-chained. Each unit waits for every other unit, so all units work as an integrated system.

All units must be on and in the RUN state before any/all will run. For example, if an out-of-limits condition stops a ramp for one unit, all other units will stop until the condition returns to within limits.

## C.2 APPLICATIONS.

Synchronization is an ideal approach for applications where you must control three or more set-points. In ECA tests, these typically would be DUT temperature, cooling air temperature, and cooling air mass flow rate.

A typical application would be an airborne package under test in a temperature-controlled environment (simulating the internal cockpit), with cooling air introduced from the outside world at a specified (controlled) temperature and velocity. Figure C-1 below shows such a typical ECA test setup.

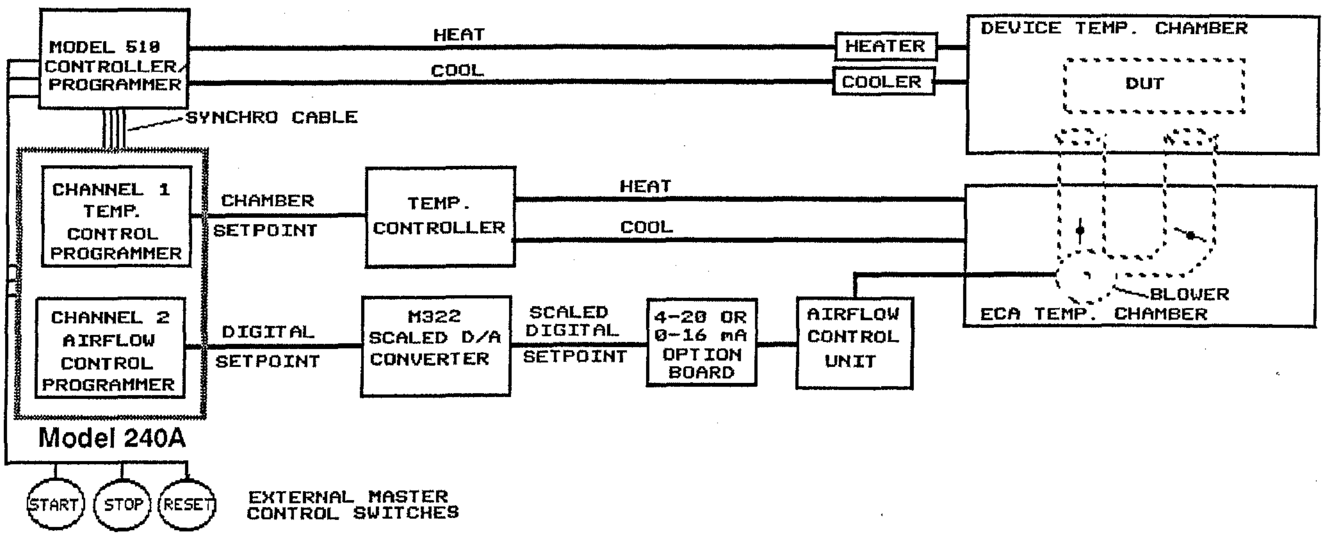


*Figure C-1. Typical Synchronizer Application.*

## C.3 INSTALLATION.

The synchronizer is a small option board installed in the programmer by JC Systems and enabled using switch S1-7. Synchronized units are connected in daisy-chain fashion with customer-installed twisted pairs between the synchronizer I/O board terminal blocks on the programmer rear panels. If no cable is connected to a programmer equipped with the synchro option board, or if S1-7 is in the off position, the programmer functions as a normal standalone type.

Synchronized units should be mounted in the same 19-in. rack or set side-by-side.

## C.4 OPERATION.

### C.4.1 Programming.

To use the synchronizer option, enter programs with the same time base and step duration for all units to be operated synchronously. If one programmer gets to the end of its step before the others, it will wait for the other programmers, and then all units will resynchronize before starting the next step. No unit will start until all units are ready.

### C.4.2 Remote Commands.

When using synchronized programmers, you don't need a computer talking to the units as a network to make the system run. External STOP, START and RESET command inputs on the individual programmers can all be connected in parallel as shown in Figure C-1. This way, one set of master switches can be used to start, stop, or reset all units simultaneously.