F80A
IEEE-488 INTERFACE BOARD

10187ML-01

MDG0395NEB3B

CHG	ECO NO	DATE	BY	CHK
G	8031	22MAR94	MG	

TABLE OF CONTENTS

ILLUSTRATIONS

1.0 GENERAL DESCRIPTION

Model F80A is an upper circuit-board option that provides an interface between the IEEE-488 (GPIB) instrumentation bus and Newport Models 204B, 258, 2003B, 2004 voltmeters and Models 267B, 268 temperature meters. One of these meters with the F80A provides a low-cost, single-range alternative to a multimeter in IEEE-488 test and control systems. By using an external power supply for the F80A, the DPM can be electrically isolated from the IEEE-488 bus.

The F80A functions as an IEEE-488 talker/listener with Service Request, Serial Poll, Device Trigger, Device Clear and Selected Device Clear capabilities. It may be programmed to send only one measurement message each time it is addressed, or it may remain a talker and send continuous measurements as each new measurement becomes available.

When addressed as a talker by the computer or other bus controller, the F80A sends the latest reading in ASCII code format to the bus controller and any other listeners on the bus. When addressed as a listener, it may be programmed by the controller to include peak, valley and average readings and/or status information with each measurement message.

The F80 compares each measurement to four programmed setpoint values. The resulting HI-LO pattern is compared to a programmed alarm pattern. A matching pattern signals an alarm by asserting the SRQ (Service Request) line of the bus. A serial poll can be conducted by the controller to locate the device creating the alarm condition.

When programmed for the Talk-Only mode, the DPM is always a talker. It can send readings to a listen-only device, such as a Newport Model 822 printer, without need for a bus controller. This is useful for data-logging applications. A delay interval may be hardwired at the external connector to select the time interval between printed readings.

Unrelated to the normal DPM functions, there are twelve digital I/O lines that may be configured as inputs or outputs in groups of four. The controller can read the logic levels of the inputs or set the logic levels of the outputs. This is a convenient means of monitoring or controlling external processes without having to add digital control devices to the bus.

1.1 THE IEEE-488 STANDARD

Since its development in 1965, IEEE-488 has emerged as the standard bus for interfacing programmable instruments to computers in the ATE and laboratory environments. Using standardized cables, it allows up to 15 instruments with unique addresses to be interfaced on a parallel bus to a central controller. High-speed TTL signals are transferred on 8 data lines and 8 control lines.

The F80A allows six of Newport's panel meters to join Newport's Model 822 panel-mount printer as full-featured instruments on the IEEE-488 bus.

1.2 ADDRESSING

Front-panel jumper connections behind the lens provide selection of the device address (0-30) that is used for both talking and listening. In normal talker-listener operation, operating parameters are downloaded from the controller to the addressed DPM using the F80A's comprehensive set of programming instructions.

1.3 OPERATING MODES

The F80A Interface can be operated in three basic modes. These provide versatility not normally found in low-cost IEEE-488 instrumentation.

1. FREE-RUN MODE: In the free-run mode the DPM takes readings continuously for purposes of display and calculation of average, peak and valley values. However, it transmits these values only when it is addressed as a talker by the controller. This relieves the controller from capturing every reading and making these calculations.

2. TRIGGERED MODE: In the triggered (HOLD) mode, no readings are taken until triggered by the controller or by external hardware. When the triggered reading is complete, a Service Request (SRQ) is sent. Upon receipt, the controller performs a serial poll to identify the requesting device and then requests the DPM reading.

   There are two trigger mechanisms:

   a) Bus trigger by the GET (Group Execute Trigger) command from the controller.

   b) Hardware trigger by a positive (False) pulse on the HOLD input.

3. TALK-ONLY MODE: The F80A can talk repeatedly to listen-only devices, such as Newport's Model 822 IEEE-488 panel-mount printer, without a bus controller connected to the system. The time interval between printouts is selected by grounding pins on the external control connector. This combination of a panel meter with F80A option and a printer creates a low-cost data logging system.

1.4 TYPES OF MEASUREMENT DATA

The F80A Interface may be programmed to provide the bus controller with the latest, average, peak and valley readings plus setpoint comparison and other status information.

1. LATEST VALUE. The transmitted data is the latest reading and is always part of the measurement message.

2. RUNNING AVERAGE. The transmitted data is software filtered by weighting the latest reading by 1 and prior readings by .9, .81, .73 etc. Noise reduction is realized without external filter hardware or data averaging by the controller.

3. PEAK VALUE. The transmitted data is the most positive reading obtained since last reset by the controller.

4. VALLEY VALUE. The transmitted data is the most negative reading obtained since last reset by the controller.

1.5 SETPOINTS PLUS ALARM MASK

The F80A Interface provides a sophisticated setpoint/alarm capability, which allows it to respond to a combination of setpoint conditions.

Four setpoints may be downloaded from the controller to the F80A. Measurements that equal or exceed these setpoints set bits in a Value Status byte. If the four-bit pattern in the Value Status byte matches the four-bit pattern in a downloaded Alarm Mask byte, the F80A signals an alarm condition by asserting the SRQ (Service Request) line of the bus. The controller can then conduct a serial poll to determine which device created the alarm condition and take appropriate action.

1.6 DIGITAL I/O LINES

The F80A Interface provides twelve logic control lines that may be configured as inputs or outputs in groups of four. These are separate from the normal DPM function. The controller can read the logic levels of the inputs and set the logic levels of the outputs. This adds digital control I/O capability to the IEEE-488 bus without adding another device or using up another bus address.

The digital inputs are a convenient means for monitoring external processes or control panel switch positions. Pull-up resistors are provided by the F80A Interface so that contact closures, opto-isolators, and open collectors can be monitored as well as TTL-level digital signals.

The digital outputs can sink up to 20 mA each, provided the total of all outputs does not exceed 80 mA. They can be used for controlling processes and indicating alarms. By driving reed relays, they can provide control for an input multiplexer connected to the DPM. By driving an external DAC, an analog signal may be generated by the controller.

For critical applications, digital output data may be echoed back to the bus controller for verification before it is transferred to the output port latches. A Control-Ready pulse advises external circuits that the transfer is complete.

1.7 INTERCONNECTION

1. PCB Connector Assembly (PA80)

   This is a DIN 1A case extension, which provides the interconnection between the F80A upper board and the DPM lower board plus I/O connections for the IEEE-488 and 12 digital I/O lines. It is required for all F80A users who do not build their own cable assembly. The back of the PA80 assembly provides a standard 24-pin female IEEE-488 connector and a double-row 20-pin PCB edge connector with solder eyelets for the 12 digital I/O lines, external power and ground.

2. Connectors (D50M and D50S)

   A 50-pin mass-termination PCB edge connector (D50M) and a 50-pin solder-eyelet PCB edge connector (D50S) are offered as options for customers who wish to build their own cable assembly. This assembly requires connectors for the IEEE-488 bus and for the 12 digital I/O lines. A PCB edge connector (D1) for the DPM lower board is also offered.

3. Ribbon-Cable Assembly

   The order of the pins on the rear connector of the F80A board allows the use of ribbon cable and mass-termination connectors for interconnection of several DPMs in close proximity to each other. A custom ribbon cable with appropriately positioned connectors may be a low-cost solution for OEM customers who build systems in a single rack.

   The ribbon cable is split for connection to the 12-line digital I/O connector and to the DPM lower board. Reasonable caution must be exercised in using unshielded ribbon cable because of noise pickup and radiation. For this reason, the cable should be short. For additional information on the ribbon-cable approach, see Appendix A.

1.8 POWER

The F80A may be powered directly from the host DPM in a non-isolated system, or from an external 5 V dc power supply in an isolated system. Maximum current required is 80 mA. With the PA80 option, the external power input is on the control I/O connector, otherwise it is applied directly to the F80A board connector.

2.0 SPECIFICATIONS

Dimensions

F80A circuit board	86 x 122 mm (W x L) (Newport DIN 1A upper board)
Length of DPM with F80A (no connector)	145 mm
PA80 PCB connector assembly (optional)	48 x 97 x 57 mm (H x W x L)
Length of DPM with PA80	173 mm

Data collection rate

Measurement rate	Determined by the DPM. 2.5 to 5 readings/s typical up to 30 readings/s with Model 258
IEEE-488 data rate	1.5 kbytes/s

Connections

F80A board	50-pin PCB edge connection without connector
PA80 PCB connector assembly option	Standard 24-pin female IEEE-488 connector, 20-pin solder-eyelet connector for 12 digital I/O lines, 50-pin PCB edge connector for F80A upper board, 36-pin D1 PCB edge connector for DPM lower board.
D50M connector option	50-pin PCB edge connector for F80A board, mass termination
D50S connector option	50-pin PCB edge connector for F80A board, solder eyelets

Power

Non-isolated operation	Power derived from DPM main board
Isolated operation	5 ± .25 V, 80 mA power required from external, isolated supply (customer-furnished)

Digital Control I/O Lines

Number of inputs	12, 8, 4, 0
Number of outputs	0, 4, 8, 12

Input level	CMOS and LSTTL compatible 2.0 to 5.5 V dc = "1" -0.5 to 0.8 V dc = "0" Iin = 0.1 mA 47 Kohm pull-up resistor provided for contact closures, opto-isolators and open collectors.
Output level	0.4 V sinking 10 mA 0.8 V sinking 20 mA 20 mA max sink current per line 80 mA max sink current for sum of all 12 lines

IEEE-488 Capabilities Implemented

SH1	Source handshake
AH1	Acceptor handshake
T5	Basic talker Serial poll Unaddress if MLA Talk-Only mode
L4	Basic Listener Unaddress if MTA Not Listen-Only mode
SR1	Service request
DT1	Device trigger
DC1	Device clear

IEEE-488 Capabilities Not Implemented

RL0	Remote/Local
PP0	Parallel poll
C0	Controller
TE0 LE0	Secondary addresses

3.0 INSTALLATION

If the F80A option is to be used with DPM models 204B, 2003B or 2004, it is necessary to add jumpers on those DPM boards to provide the signals required by the F80A board. If the F80A option is ordered with one of these DPM's and is received with both the main DPM board and the F80A board installed in a case, then the jumpers have been added by the factory. However, if the F80A board is separately received for field installation with a DPM, the instructions in Sections 3.1 and 3.2 must be executed. These sections may be skipped for DPM models 258, 267B and 268 because the required signals are available from the DPM without the addition of jumpers.

3.1 FIELD INSTALLATION

1. Remove AC power from the instrument to which the F80A option is to be installed.
2. Remove lens, two screws, clamp rings and slide retainers shown in outline and mounting drawings for the DPM.
3. Remove top half of case.
4. Carefully remove bottom board from case by pressing on board retainers.
5. Locate appropriate circuit points on bottom DPM board as listed in Figure 3-1 and install insulated #24 wires as required.
6. Reassemble lower board into lower case half.
7. Install F80A option board into upper-case half and reassemble meter.

   NOTE: Before reassembling the meter, read Section 3.6 and either Section 4.1 or 4.2, Item 2 so the appropriate power jumpers can be installed on the F80A board while it is out of the case.

8. Install the PA80 connector assembly. If this option not ordered, then wire the F80A connector to the DPM connector according to wire list in Figure 3-7.

3.2 DPM BOARD JUMPERS

The following jumpers must be installed on the DPM lower board.

Model No.	Signal	Jumper	#24AWG Insulated Wire
204B	Reference	E16 to E34	5"
2003B	Reference	P3-5 to E13	7"
2004	+5 V Reference	P3-8 to E14 P3-5 to E13	7" 7"
267B	-	None required	-
268	-	None required	-
258	-	None required	-

Figure 3-1 DPM Jumper Requirements for F80A Option

3.3 DPM AND F80A INTERCONNECTIONS

There are three alternatives for system connections to the F80A Interface. These are the PA80 Connector Assembly, an individual wire cable assembly and a ribbon cable assembly. The PA80 connector assembly is available as an option from the factory while the other two can be wired by the customer according to the wiring tables and instructions that follow.

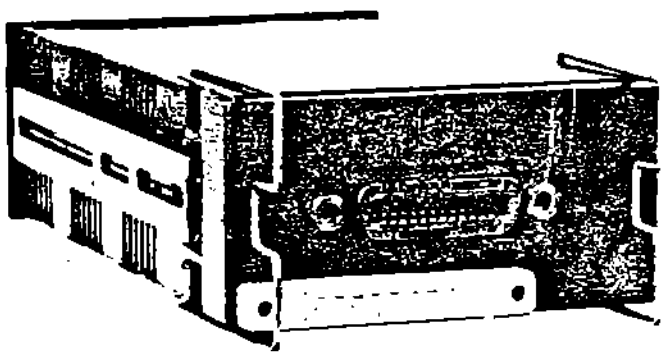


Figure 3-2 PA80 Connector Assembly Attached to DPM

A. PA80 Connector Assembly

If the PA80 Connector Assembly is ordered, all connections between the DPM and the F80A are made via PC traces on a rigid PCB assembly. Also, connections are made between the F80A and the IEEE-488 connector and the Control I/O Connector.

Since the DPM models have different pinouts, it is necessary to use solder-bridge jumpers to configure the PA80 for the particular DPM with which it is used. If the PA80 option is ordered with a particular model DPM, the appropriate solder-bridge jumpers should be in place. If not, or if a decimal point is required, refer to Figure 3-4 to identify the proper jumper locations. The DPM manual identifies the significance of the decimal point nomenclature.

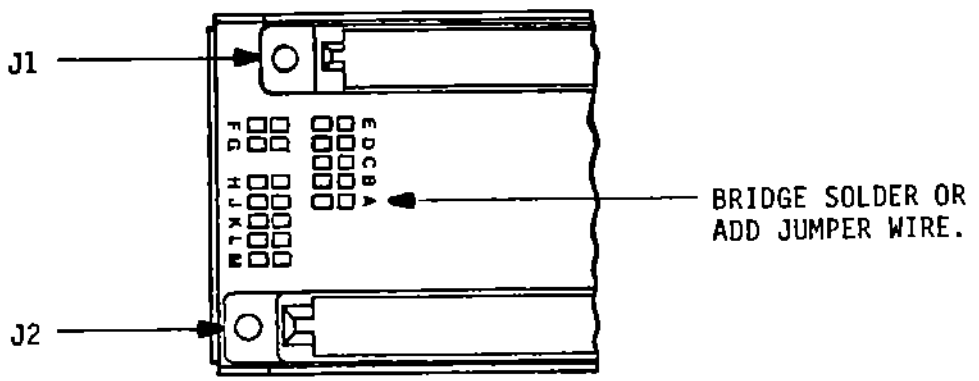


Figure 3-3 PA80 Solder-Bridge Jumper Locations

DPM MODEL	BIT 1		REF			+5V		DEC 1	DEC 2	DEC 3	DEC 'A'	DEC 'B'
	F	G	J	K	H	L	M	E	D	C	A	B
2003B	—	●	●	—	—	●	—	—	—	—	◁▷ OR ◁▷	
2004	—	●	●	—	—	—	●	—	—	—	◁▷ OR ◁▷	
204B	●	—	●	—	—	●	—	◁▷ OR ◁▷ OR ◁▷			—	—
267B	●	—	—	●	—	●	—	—	—	—	—	—
268	●	—	—	●	—	●	—	—	—	—	—	—
258	●	—	—	—	●	●	—	◁▷ OR ◁▷ OR ◁▷			—	—

● = CLOSE ◁▷ = OPEN

Figure 3-4 PA80 Solder-Bridge Jumper Connections

After the solder-bridge jumpers have been made, push the PA80 connector assembly onto the DPM and F80A card-edges. Place the PA80 adapter cover over the PA80 assembly and secure with the retainers (Figure 3-2).

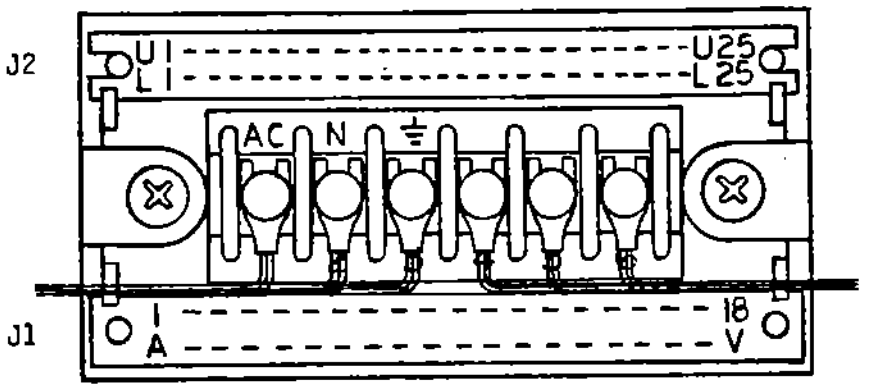


Figure 3-5 DPM and F80A Connector Pin Identification

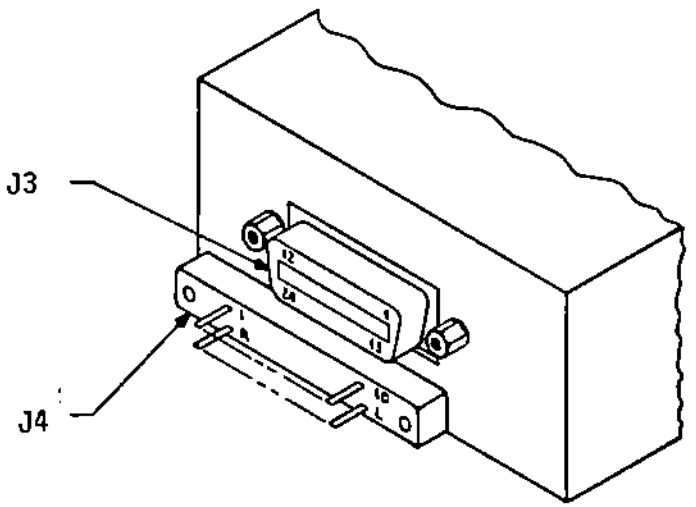


Figure 3-6 PA80 Rear Connector Pin Identification

B. Individual Wire Cable Assembly

Without the PA80 Connector Assembly, the DPM and F80A boards can be interconnected with individual wires using a 50-pin PCB edge connector with solder eyelets (option D50S) to connect to the F80A board according to the wire list of Figure 3-7. Then individual wires can be used to connect to both an IEEE-488 connector and a control I/O connector according to the wire lists of Figures 3-8 and 3-9.

Suggested connectors with solder-eyelets or posts for constructing an individual wire cable assembly are:

Connector	Description	Mfg	Mfgs P/N
J1	36-pin PC edge	SAE	SCU 18D/1-2
J2	50-pin PC edge	KEL-AM	RC25-0142-5
J3	24 pin IEEE-488	TRW CINCH	57-20240
J4	25 pin D-subminiature (See Section 3.5)	CANNON	DB-25S

F80 Upper Board Connector	DPM Connector Pin# 204B	2003B	2004	267B	268	248	Ribbon Cable Wire Color	Ribbon Cable Wire #	Signal Name
J2-L22	J1-N	J1-N	J1-N	J1-N	J1-N	J1-N	Orange	43	+ POLARITY
J2-U22	J1-R	J1-R	J1-R	J1-R	J1-R	J1-R	Yellow	44	HOLD
J2-L23	J1-P	J1-P	J1-P	J1-P	J1-P	J1-P	Green	45	DATA READY
J2-U23	J1-D	J1-7	J1-7	J1-D	J1-D	J1-D	Blue	46	COUNTER BIT 1
J2-L24	-	-	-	-	-	-	-----	--	----
J2-U24	J1-U	J1-U	J1-U	J1-V	J1-V	J1-S	Gray	48	REFERENCE
J2-L25	J1-T	J1-T	J1-T	J1-T	J1-T	J1-T	White	49	DIGITAL GND
J2-U25	J1-15	J1-15	J1-18	J1-15	J1-15	J1-15	Black	50	+5 V OUT

Figure 3-7 DPM to F80A Wire List

C. Ribbon Cable Assembly

A third alternative for interconnection is to use ribbon cable with a 50-pin mass-termination connector (option D50M) to connect to the F80A board. The order of pins on the F80A board was carefully chosen to allow mass termination connectors to be used at the other end of the ribbon cable for both the IEEE-488 and control I/O connections.

One advantage of the ribbon cable assembly is that the IEEE-488 outputs of several DPMs can be connected in parallel on the ribbon cable with only one IEEE-488 connector at the end of the assembly to connect to the IEEE-488 bus. This eliminates the cost of several expensive IEEE-488 cables to connect each DPM to the bus, making it attractive to OEM customers that build systems in a single rack or cabinet. However, caution must be exercised in using unshielded ribbon cable because of noise pickup and radiation.

A portion of the ribbon cable needs to be split off for connection to the DPM main board and another portion for connection to the control I/O connector.

Details for the construction of a ribbon cable assembly appear in Appendix A, including recommended connector types.

3.4 IEEE-488 BUS WIRING

F80A Upper Board Connector	Ribbon Cable Wire Color	Cable Wire #	IEEE-488 Connector
J2-L1	Brown	1	24 Gnd, LOGIC
J2-U1	Red	2	12 SHIELD
J2-L2	Orange	3	23 Gnd, (11)
J2-U2	Yellow	4	11 ATN
J2-L3	Green	5	22 Gnd, (10)
J2-U3	Blue	6	10 SRQ
J2-L4	Violet	7	21 Gnd, (9)
J2-U4	Gray	8	9 IFC
J2-L5	White	9	20 Gnd, (8)
J2-U5	Black	10	8 NDAC
J2-L6	Brown	11	19 Gnd, (7)
J2-U6	Red	12	7 NRFD
J2-L7	Orange	13	18 Gnd, (6)
J2-U7	Yellow	14	6 DAV
J2-L8	Green	15	17 REN
J2-U8	Blue	16	5 EOI
J2-L9	Violet	17	16 DIO8
J2-U9	Gray	18	4 DIO4
J2-L10	White	19	15 DIO7
J2-U10	Black	20	3 DIO3
J2-L11	Brown	21	14 DIO6
J2-U11	Red	22	2 DIO2
J2-L12	Orange	23	13 DIO5
J2-U12	Yellow	24	1 DIO1

Figure 3-8 F80A to IEEE-488 Bus Wire List

3.5 CONTROL I/O WIRING

The Control I/O connector is furnished with the PA80 connector assembly and is a 20-pin PC edge connector. The Control I/O connector for the two customer-wired cable assemblies can be selected by the customer. A suggestion is one of the following female 25-pin D subminiature connectors. Pinouts appear in Figure 3-9.

Individual connections	CANNON	DB-25S
Mass-terminated connections	AMP	745242-2

F80A Upper Board Connector	Cable Wire Color	Ribbon Cable Wire #	Control I/O Connector: Pin ID	Control I/O Connector: PA80 20-Pin PC Edge	Control I/O Connector: 25-Pin D Subminiature
J2-L13	Green	25	C1	1	13
J2-U13	Blue	26	C2	A	25
J2-L14	Violet	27	C3	2	12
J2-U14	Gray	28	C4	B	24
J2-L15	White	29	C5	3	11
J2-U15	Black	30	C6	C	23
J2-L16	Brown	31	C7	4	10
J2-U16	Red	32	C8	D	22
J2-L17	Orange	33	C9	5	9
J2-U17	Yellow	34	C10	E	21
J2-L18	Green	35	C11	6	8
J2-U18	Blue	36	C12	F	20
J2-L19	Violet	37	GND B	9,10	7
J2-U19	Gray	38	Control Ready	8, J	19
J2-L20	White	39	GND B	K, L	6
J2-U20	Black	40	+5 V B	7, H	18

Figure 3-9 F80A to Control I/O Wire List

3.6 POWER CONNECTIONS

The Model F80A Interface can provide electrical isolation between the DPM circuitry and the IEEE-488 bus circuitry. This reduces common mode errors at the low level DPM input that otherwise could result from common ground currents in the system.

The supply voltage for the circuitry on the DPM side of the barrier is supplied by the DPM and is labeled +5 V A.

Operating in an isolated mode, the supply voltage for the circuitry on the bus side of the barrier must be supplied by an external supply and is labeled +5 V B.

The current requirements are:

+5 V A from DPM = 15 mA
+5 V B from external supply = 80 mA

As an alternative, the F80A can operate in a nonisolated mode which does not require an external power supply. Instead, the power is furnished from the DPM by jumpering the +5 V A line to the +5 V B line and GND A to GND B.

1. Nonisolated Mode

The normal configuration for shipment is the DPM-powered, nonisolated mode. This allows an initial test to be made without an external power supply.

This mode requires jumpers E4/E5 and E2/E3 on the PA80 assembly (Figure 3-10) if present, or on the F80A board (Figure 3-11) if the PA80 assembly is not used.

2. Isolated Mode

To configure the isolated mode, remove the jumpers E4/E5 and E2/E3 from the PA80 assembly (Figure 3-12) or the F80A board (Figure 3-13). External power is then applied according to Figure 3-12 if a PA80 assembly is present or Figure 3-13 if it is not used.

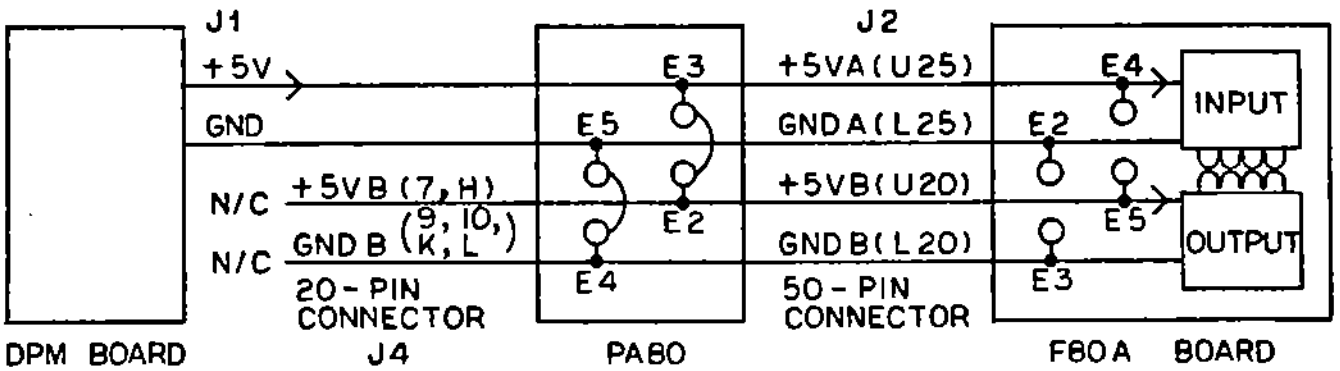


Figure 3-10 DPM Power - F80A With PA80

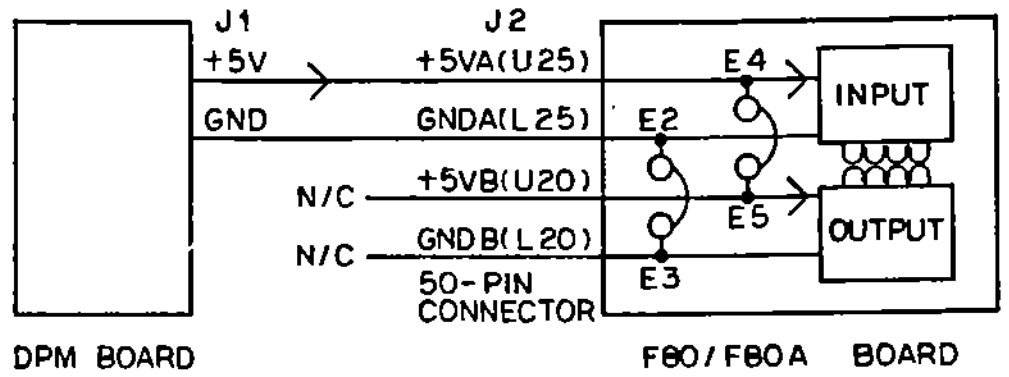


Figure 3-11 DPM Power - F80/F80A Without PA80

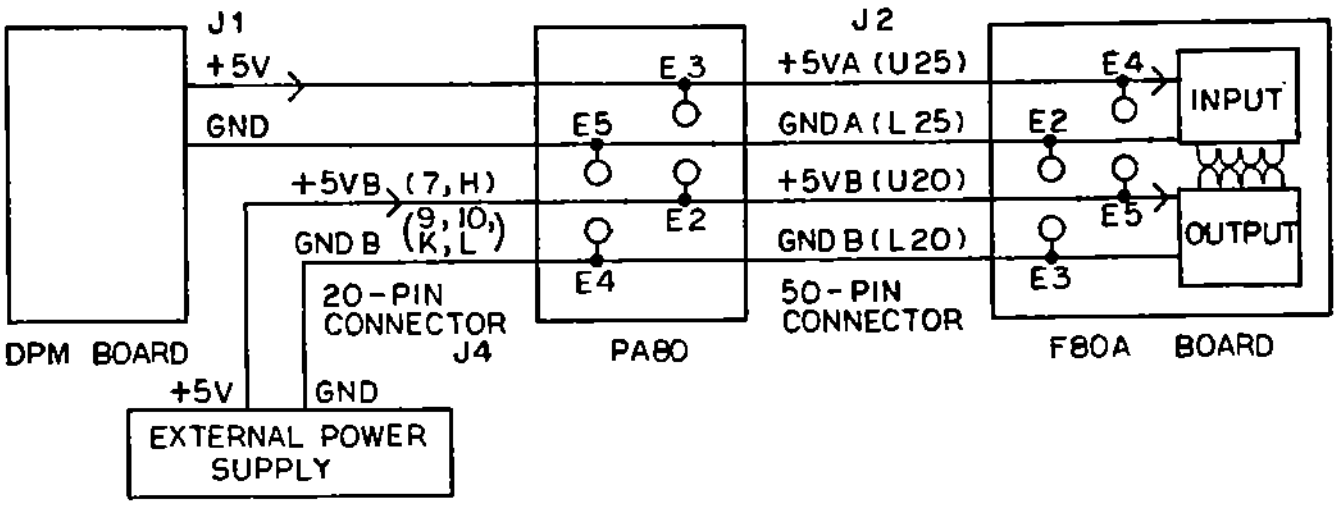


Figure 3-12 Isolated Power - F80A With PA80

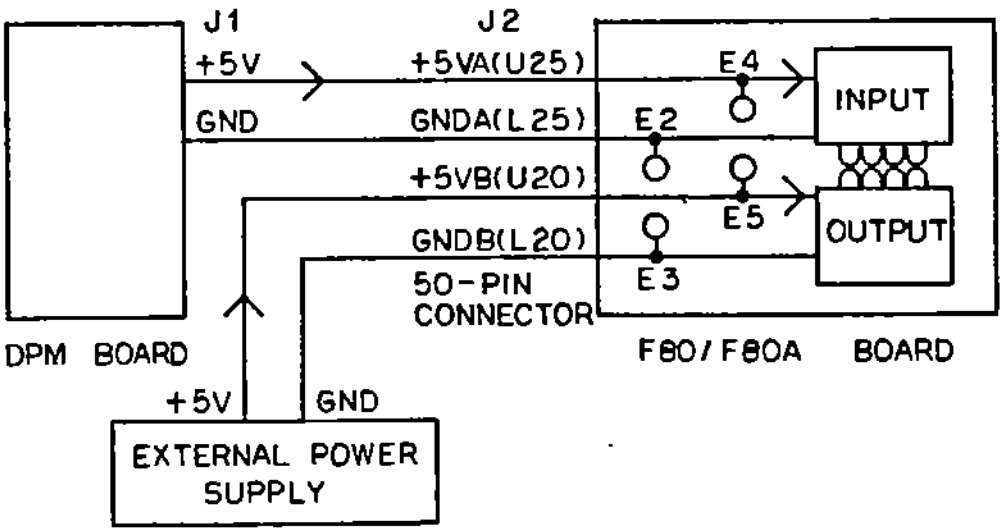


Figure 3-13 Isolated Power - F80/F80A Without PA80

4.0 STARTUP

The purpose of this section is guide the user in the process of receiving a DPM measurement at the controller over the IEEE bus. Once this has been achieved, it is probable that the interconnections and jumpers are correct. Then it's only a matter of learning and applying the program instructions and data formats from Sections 5 to 12.

When power is applied, the F80A assumes the default conditions of the program instructions (those with an * in Section 5.2). This provides the capability of sending a measurement to the controller without the controller first sending program instructions to the DPM. Therefore, it is advantageous to achieve success in receiving a correct measurement from the DPM before attempting to program it.

It is assumed that this initial test will be performed in the DPM-powered, nonisolated mode. This should be the jumpered mode of the DPM when received (see Section 3.6). Since it is nonisolated, care must be taken to ensure that the DPM input leads are not connected to a high common mode voltage with respect to the IEEE-488 bus. This is easily achieved by shorting the DPM signal input leads together and letting them float with respect to IEEE-488 ground.

Appendix B contains the listing of a startup program for an IBM PC controller using a National Instruments IEEE-488 interface card. This program is known to work with the hardware involved, which eliminates one of the possible error sources in achieving initial startup. If the users IEEE-488 interface card is from some other manufacturer, the startup program can still be used as a guide for writing a startup program using different nomenclature.

If the meter fails to return a measurement value to the controller when addressed as a talker, there are two checklists to follow. One is for a DPM/F80A unit with a PA80 connector assembly attached and the other for a customer-furnished cable assembly in place of the PA80.

4.1 CHECKLIST USING PA80 CONNECTOR ASSEMBLY

With DPM input leads shorted and floating, apply AC line power.

1. DPM display should light.
   If not, check AC line connections to DPM.
2. +5 V should be present on pin 7 of J4, the 20-pin control I/O connector (Figure 3-6).

   If not, remove the PA80 cover (Figure 3-2) and verify that jumpers E4/E5 and E2/E3 are present (Section 3.6). If they are and there is no voltage on E23 (+5 V A) and the DPM is a 2004, the DPM internal jumpers may be missing (Section 3.2).

3. Remove the front lens of the DPM and verify that the jumper plug pattern agrees with address 7 used by the startup program (Figure 5-2). Address 7 is normally jumper-programmed by the factory.

4. Remove the PA80 cover and verify that the solder-bridge jumpers A to M correspond to the proper DPM model number (Section 3.3A).

5. Test pin J2-U22 for a high (false) HOLD signal. If low true, try turning the power off and on.

6. Use an oscilloscope to observe the DPM REFERENCE signal, pin J2-U24. It should alternate between ground and a high logic level at the DPM reading rate.

   If not, and the DPM is a 204B, 2003B or 2004, the DPM internal jumpers may be missing (Section 3.2).

7. Use an oscilloscope to observe the BIT 1 signal, J2-U23. It should be a burst of pulses repeating at the DPM read rate.

8. Remove the F80A board from the case and verify that shorting-block jumpers are in locations 6B, 7A, 8B and locations 1-5 are without jumpers.

   Note 1: If the reading returned to the controller is half the DPM displayed reading, it is probable that the location 8 jumper is in position A instead of position B.

   Note 2: If address jumpers are in locations 1-5 instead of on the vertical pin-forest jumper board behind the lens, verify that the jumpered address agrees with the Startup program (address 7 is 1A, 2A, 3A, 4B, 5B).

9. Use an IEEE-488 Bus Analyzer to ensure that the STARTUP program is providing the correct bus signals.

4.2 CHECKLIST USING CUSTOMER CABLE ASSEMBLY

With DPM input leads shorted and floating, apply AC line power.

1. DPM display should light.
   If not, check AC line connections to DPM.

2. +5 V should be present on pin J2-U20 (Figure 3-5).

   If not, remove the F80A board from the case and verify that jumpers E4/E5 and E2/E3 are present (Section 3.6). Return the board to the case and check the voltage on pin J2-U25. If not present and the DPM is a 2004, the DPM internal jumpers may be missing (Section 3.2).

3. Remove the front lens of the DPM and verify that the jumper plug pattern agrees with address 7 used by the STARTUP program (Figure 5-1). Address 7 is normally jumper-programmed by the factory.

4. Verify that the interconnections between the DPM and the F80A are correct for the DPM model number (Figure 3-7).

   Verify that the interconnections between the F80A and the IEEE-488 connector are correct (Figure 3-8).

5. Test pin J2-U22 for a high (false) HOLD signal. If low true, try turning the power off and on.

6. Use an oscilloscope to observe the DPM REFERENCE signal, pin J2-U24. It should alternate between ground and a high logic level at the DPM reading rate.

   If not, and the DPM is a 204B, 2003B or 2004, the DPM internal jumpers may be missing (Section 3.2).

7. Use an oscilloscope to observe the BIT 1 signal, J2-U23. It should be a burst of pulses repeating at the DPM read rate.

8. Remove the F80A board from the case and verify that shorting-block jumpers are in locations 6B, 7A, 8B and locations 1-5 are without jumpers.

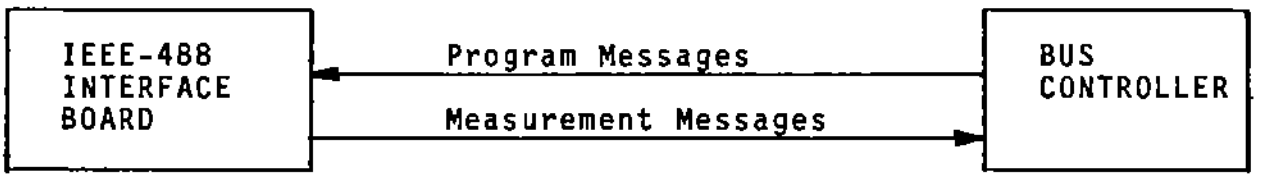
   Note 1: If the reading returned to the controller is half the DPM displayed reading, it is probable that the location 8 jumper is in position A instead of position B.

   Note 2: If address jumpers are in locations 1-5 instead of on the vertical pin-forest jumper board behind the lens, verify that the jumpered address agrees with the STARTUP program (address 7 is 1A, 2A, 3A, 4B, 5B).

9. Use an IEEE-488 Bus Analyzer to ensure that the STARTUP program is providing the correct bus signals.

5.0 BASIC OPERATION

The Interface board may be addressed as a talker or a listener. As a talker, it sends measurement values to the bus and as a listener it receives program instructions from the bus controller.



5.1 PROGRAM MESSAGES

Each program message unit or instruction may be sent alone or combined in a continuous string with other program message units. Each message unit consists of an alpha header by itself or followed by a data string. No separator is necessary between program message units. To meet ANSI/IEEE Std 728-1982 (IEEE Recommended Practice for Code and Format Conventions), quote marks should be placed before and after the data string. However, since the Interface ignores all quote marks, they may be omitted.

Example: Send Value Status byte and Average value.

H"1"J"1" or H1J1

All program message units are processed as they are received. No end terminator is required. The unlisten command should be sent following the program message to prevent other data on the bus from being interpreted as program instructions. The next section provides a summary of the program instructions. Details of each instruction are found in the sections following.

There are two kinds of program instructions, stored and demand.

Stored instructions determine the nature of each measurement message received from the DPM and remain in effect until changed by another stored instruction.

Demand instructions are in effect for one measurement message only and must be repeated each time a demand message is desired. A demand instruction takes precedence over a stored instruction.

Stored and demand instructions may be sent in the same program message string. Stored instructions will be incorporated as they are received. If more than one demand instruction is received, only the last one will be recognized.

The next measurement message will be the demand message, followed by stored messages when addressed to talk again.

5.2 PROGRAM INSTRUCTION SUMMARY

* indicates power-on reset (default) value
\# represents an alphanumeric character

STORED INSTRUCTIONS

Reference	Instruction	Description
6.4	N0	Do not send carriage return with each measurement message unit.
6.4	N1	* Send carriage return with each measurement message unit.
6.4	O0	* Do not send line feed with each measurement message unit.
6.4	O1	Send line feed with each measurement message unit.
7.1	H0	* Do not send Value Status byte with reading.
7.1	H1	Send Value Status byte with reading.
7.2, 7.3	I0	* Do not send System & Mode Status bytes with reading.
7.2, 7.3	I1	Send System & Mode Status bytes with reading.
6.6	J0	* Do not send Average value with reading.
6.6	J1	Send Average value with reading.
6.7	K0	* Do not send Peak & Valley values with reading.
6.7	K1	Send Peak & Valley values with reading.
6.3	Y#	# = 0 Do not send decimal point. # = 1-7 Send decimal point in position # .
8.1	P#######	Setpoint A value
8.1	Q#######	Setpoint B value
8.1	R#######	Setpoint C value
8.1	S#######	Setpoint D value
8.2	V#	Alarm Mask
11.1	T#	External Control I/O configuration
11.3	Z###	External Control Output
9.0	L0	* Free-Run Mode
9.0	L1	Triggered Mode
6.5	M0	* Send continual readings.
6.5	M1	Send one reading only, each time addressed.
6.6, 8.1	U0	* Use Latest value for setpoint comparison.
6.6, 8.1	U1	Use Average value for setpoint comparison.

6.7	A	Reset Peak value.
6.7	B	Reset Valley value.
6.7	C	Reset both Peak and Valley values.
12.3	E	Power-On Reset.
11.2	D	Read External Control inputs.
11.3	F	Transfer External Control outputs to output drivers.
	X#	# = 0 Send Setpoint A value
		= 1 Send Setpoint B value
		= 2 Send Setpoint C value
		= 3 Send Setpoint D value
		= 4 Send Latest value
		= 5 Send Average value
		= 6 Send Peak value
		= 7 Send Valley value
		= 8 Send Alarm Mask
		= 9 Send Value Status byte
		= : Send System Status byte
		= ; Send Mode Status byte
		= < Send IEEE Status byte
		= ? Send Control Output Buffer

5.3 MEASUREMENT MESSAGES

Each measurement message consists of one or more measurement units. These measurement units may be readings, status bytes, control inputs or echoed program instructions. Each measurement unit in a measurement message is separated with a carriage return, a line feed or both. Readings and echoed setpoints are numeric values and may appear in the measurement message with or without a decimal point.

Status bytes and Control I/O values are ASCII strings and are enclosed in quotes if the separator contains a line feed (LF or CR/LF). This is to meet ANSI/IEEE Std 728-1982 (IEEE Recommended Practice for Code and Format Conventions). Each measurement message terminates with EOI asserted coincident with the last character.

Example:

	Value Status Byte	Latest Value	Average Value
CR separator	42(CR)+0015.25(CR)+0014.78(CR)		
			EOI
LF separator	"42"(LF)+0015.25(LF)+0014.78(LF)		
			EOI

NOTE: When interfacing with Hewlett Packard Controllers, send a line feed character with each measurement message unit (stored instruction 01, where 0 is the letter).

5.4 TALK/LISTEN ADDRESSING

The talk and listen addresses are the same and are set by jumper plugs on the address array board behind the front panel or on the F80A main board. These addresses may be any decimal number from 0 to 30. Address 31 is reserved for the untalk and unlisten commands and cannot be used for a DPM address. The F80A will ignore any Secondary Address sent by the controller.

ADDRESS ARRAY BOARD

This is a small vertical board that is soldered at a right angles to the main F80A board at the front. When the F80A board is installed with a meter in a case, the address array board appears behind the cutout in the meter display board. It may be accessed by removing the lens from the front of the case. This allows the meter talk and listen address to be changed without removing the installed meter from the panel.

There are five jumper plugs furnished for addressing but not all may be active for a given address. The inactive jumpers may be removed or may be put in inactive positions for safekeeping. Figure 5-1 shows the positions of the jumpers for all addresses from 0 to 30. The dark jumpers are the active ones and are required. The light jumpers are the inactive ones and may be removed or left in the positions shown. A pair of needle-nose pliers are recommended for removing and replacing the jumpers.

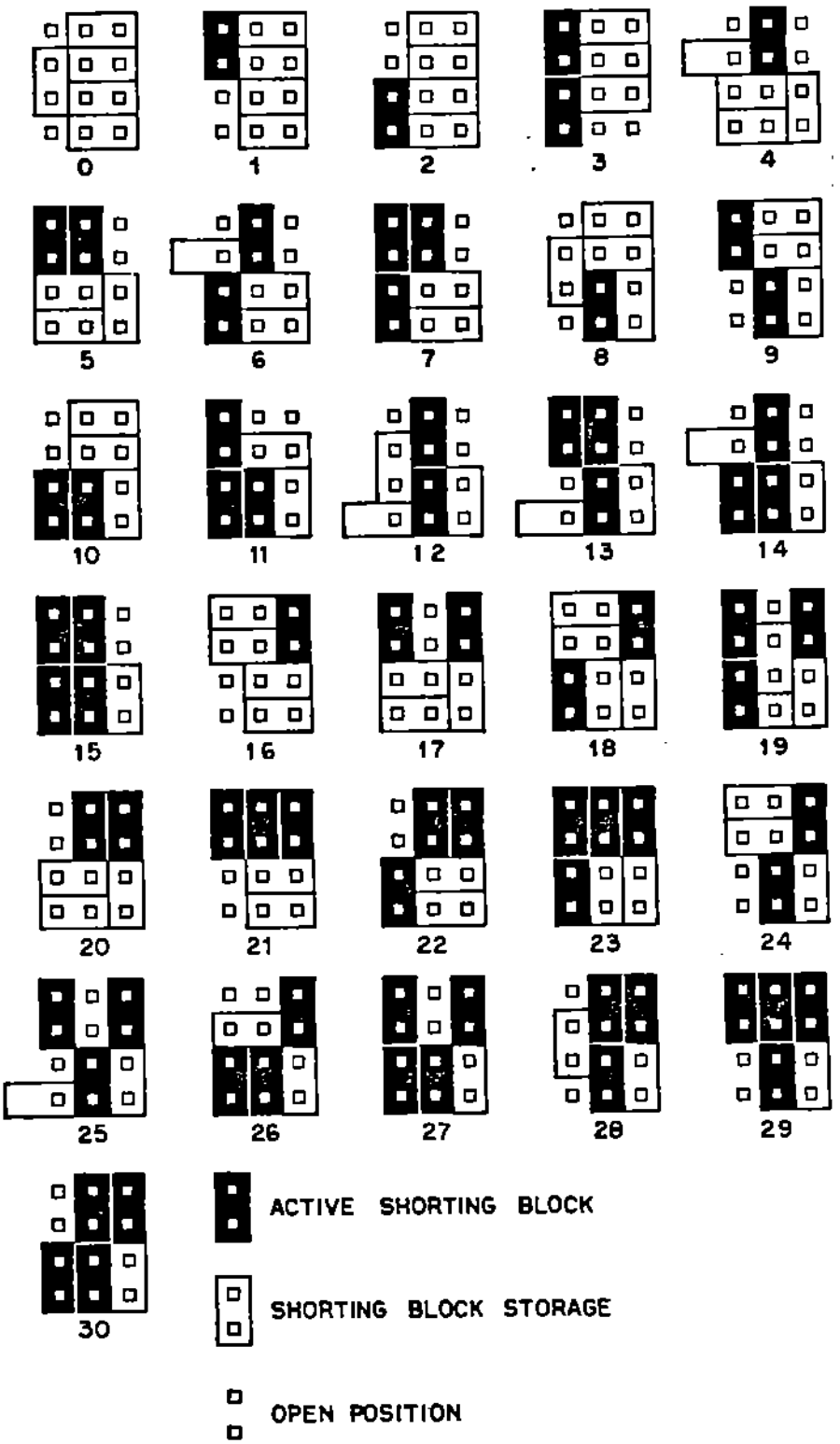


Figure 5-1 Address Programming - Address Array Board

F80A BOARD

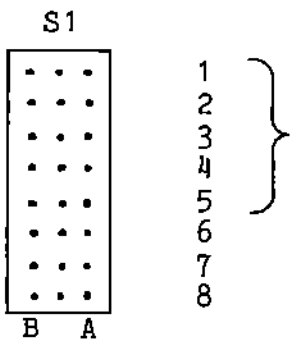
As an alternative to programming the talk and listen address on the front panel address array board, parallel connections are provided on the F80A main board. Five active and inactive jumper positions appear in a row and may be programmed according to the following table. It is necessary to remove the F80A board from its case to exercise this alternative and while inconvenient, this requirement provides a measure of security against an undesired change of address at the front panel.

TALK AND LISTEN ADDRESSES

DECIMAL ADDRESS	F80A MAIN BOARD JUMPER POSITION 5	4	3	2	1	DECIMAL ADDRESS	F80A MAIN BOARD JUMPER POSITION 5	4	3	2	1
0	0	0	0	0	0	16	1	0	0	0	0
1	0	0	0	0	1	17	1	0	0	0	1
2	0	0	0	1	0	18	1	0	0	1	0
3	0	0	0	1	1	19	1	0	0	1	1
4	0	0	1	0	0	20	1	0	1	0	0
5	0	0	1	0	1	21	1	0	1	0	1
6	0	0	1	1	0	22	1	0	1	1	0
7	0	0	1	1	1	23	1	0	1	1	1
8	0	1	0	0	0	24	1	1	0	0	0
9	0	1	0	0	1	25	1	1	0	0	1
10	0	1	0	1	0	26	1	1	0	1	0
11	0	1	0	1	1	27	1	1	0	1	1
12	0	1	1	0	0	28	1	1	1	0	0
13	0	1	1	0	1	29	1	1	1	0	1
14	0	1	1	1	0	30	1	1	1	1	0
15	0	1	1	1	1	NOT ALLOWED	1	1	1	1	1

"1" is the A position
"0" is the B position

Figure 5-2 Address Programming - F80A Board



Address (Figure 5-2)

Talk Only (Section 13.0)
Zero Suppression (Section 6.2)
Reserved (Omit or B position only)

Position A = active Position B or none = inactive

Figure 5-3 Program Jumper Locations

6.0 DPM MEASUREMENTS

When addressed as a talker, the F80A sends the latest DPM measurement value. This value will agree exactly with the DPM display. If no program instructions have been sent since the power-on reset, the default conditions exist which provide the latest measurement value terminated with a Carriage Return and the assertion of the EOI line.

The F80A may be programmed by the bus controller to include status bytes and Average, Peak and Valley Values in the measurement message. If present, these always appear in the same order regardless of the order programmed. They are sent in one continuous string of characters but each message unit has its own terminator. If this terminator includes a carriage return, the string prints as a list. Any message units not present have their position in the string omitted. Knowledge of this order allows the computer controller to process the data.

* Optional	Format	Description	Program Instruction
*	XX CR	Value Status byte	H1
*	XX CR	System Status byte	I1
	XX CR	Mode Status byte	
	+XXXXXX CR	Latest Value	Always included
*	+XXXXXX CR	Average Value	J1
*	+XXXXXX CR	Peak Value	K1
	+XXXXXX CR	Valley Value	

Figure 6-1 Message Unit Order

6.1 DPM READING FORMAT

DPM readings may be sent with the following format:

+001.234(CR)(LF)

This applies to the Latest, Average, Peak and Valley Values. The leading zeros, decimal point and terminators are optional. The default format is a sign, six digits and a Carriage Return.

Example of default format:

+001.234(CR)

6.2 ZERO SUPPRESSION

Jumper position 7 on the F80A is set to the ON (A) position to provide leading zero suppression of the DPM readings. It is applied only to measurement message units resulting from Stored Instructions.

Examples of readings with zero suppression:

+23 -1.23 +0 -0.014

6.3 DECIMAL POINTS

The default condition provides no decimal point for the DPM reading. A decimal point may be programmed with the Y instruction followed by a digit from 1 to 7. The digit signifies the location of the decimal point. It applies only to measurement message units resulting from Stored Instructions.

.9.9.9.9.9.9.

Y 7 6 5 4 3 2 1 0 = no decimal point

Examples: Y2 +01234.5
Y5 +01.2345

6.4 TERMINATORS

A choice of terminators (and separators for multiple measurement message units) is available through Stored Instructions.

NO No Carriage Return
* N1 Carriage Return

* O0 No Line Feed (letter O, number 0)
O1 Line Feed (letter O, number 1)

Combinations

N000 No terminator/separator
N100 Carriage return only
N001 Line feed only
N101 Carriage return & line feed

One of the last two is required as a separator by the ANSI/IEEE std 728-1982. The EOI line is always asserted with the final byte of the measurement message. The termination choice applies to all measurement message units.

6.5 SEND CONTINUAL/ONCE MODES

The Interface is in the Send Continual mode following a power-on reset. Measurement values are sent continually as long as there is

a listener on the bus. An output buffer is filled with the next measurement value following the transmission of the last value. If the Interface is unaddressed as a talker, that value remains in the buffer and is transmitted the next time the Interface is addressed as a talker. Since this could be old data, there are three alternatives to this undesirable condition.

1. Have the controller disregard the first reading received after addressing the Interface as a talker.

2. Have the controller send a DCL (Device Clear) or SDC (Selected Device Clear) command to clear the output buffer before addressing the Interface as a talker.

3. Send either the Value Status byte or the System and Mode Status bytes with the latest reading. The output buffer is updated every measurement in this format.

4. Have the controller put the Interface in the Send Once mode. Only one measurement value will be sent each time the Interface is addressed as a talker. The buffer will not be filled with the next value following the transmission of the last value.

    * M0 = Send Continual
      M1 = Send Once

6.6 AVERAGE DPM READINGS

A running (weighted) average is calculated with every reading of the DPM, even though the DPM may not be sending readings over the bus for extended periods of time. If programmed with a J1, this Average Value is sent with the Latest Value whenever a measurement message is requested by the controller.

Each time the DPM makes a conversion, a new running average is calculated by adding .1 of the new reading to .9 of the old average. In response to a step input, this results in a time constant (63.7% of final value) of about 10 conversion periods. If the reading rate is 4/s (.25 s/reading), the time constant would be about 2.5 s.

The Average Value should be used when there is a large noise component to the DPM input signal and speed of response can be sacrificed for filtering benefits.

Normally, continual setpoint comparisons are made using the Latest reading. However, with noisy input signals, it may be advantageous to make the setpoint comparisons using the Average Value. This could prevent the premature setting of the alarm bit associated with the four setpoint values. This choice can be implemented with the program instruction U1.

6.7 PEAK AND VALLEY DPM READINGS

Every reading of the DPM is captured and compared to the stored Peak and Valley Values. If the Latest reading is more positive than the Peak Value, it becomes the new Peak Value. Likewise, if the Latest

reading is more negative than the Valley Value, it becomes the new Valley Value. Whenever a new peak or valley value is received, the New Peak or New Valley bit is set in the Value Status byte. By observing the Value Status byte on a regular basis, it can be determined when a new Peak and Valley Value has been received and a request made for that value to be transmitted. After the Value Status byte has been sent and received by the controller, the New Peak and New Valley bits are reset. They are also reset when the Peak and Valley Values are transmitted. The Peak and Valley Values will be sent with every measurement message if the stored program instruction K1 is sent to the DPM. The Peak and Valley Values will be sent on request with the demand program instructions X6 and X7, respectively.

The Peak and Valley Values may be reset individually with the demand program instructions A and B, respectively, or simultaneously with the demand program instruction C.

7.0 THE NIBBLE FORMAT

All device Status bytes and Control I/O data are passed through the bus in the Nibble format of 4 bits per data byte. Each Status byte is split into 2 Nibbles and the 12 I/O lines are divided into 3 Nibbles.

Each Nibble is converted to an ASCII character by adding the four most significant bits 0011, representing the numerical column of the ASCII code chart. ASCII characters are conveniently sent and received by computer controllers. When two or three Nibbles are transmitted on the bus, the ASCII character containing the most-significant Nibble is sent first and the ASCII character containing the least-significant Nibble is sent last.

ASCII Column				Nibble				ASCII
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Character
0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	1	1
0	0	1	1	0	0	1	0	2
0	0	1	1	0	0	1	1	3
0	0	1	1	0	1	0	0	4
0	0	1	1	0	1	0	1	5
0	0	1	1	0	1	1	0	6
0	0	1	1	0	1	1	1	7
0	0	1	1	1	0	0	0	8
0	0	1	1	1	0	0	1	9
0	0	1	1	1	0	1	0	:
0	0	1	1	1	0	1	1	;
0	0	1	1	1	1	0	0	<
0	0	1	1	1	1	0	1	=
0	0	1	1	1	1	1	0	>
0	0	1	1	1	1	1	1	?

The only exception to the use of the Nibble format is the IEEE-488 Status byte. The seven least-significant bits are sent as an ASCII character spanning the full range of the ASCII code chart. The eighth bit is a spare and has no significance.

7.1 VALUE STATUS BYTE

The Value Status byte is divided into two nibbles of data and transmitted over the bus in the Nibble format, most-significant nibble first.

	Bit #	Value Status Byte
Most Significant Nibble	7	Equal or more positive than Setpoint D
	6	Equal or more positive than Setpoint C
	5	Equal or more positive than Setpoint B
	4	Equal or more positive than Setpoint A
Least Significant Nibble	3	Spare
	2	Listen Error
	1	New Valley
	0	New Peak

7.1.1 Most-Significant Nibble

The four bits of the Most-Significant Nibble represent the status of the most recent Setpoint comparisons with either the Latest Value (program instruction U0) or the Average Value (program instruction U1). These comparisons are made and the status bits updated following each DPM reading. After the four Setpoint bits are updated, they are compared to the corresponding four bits of the Alarm Mask (see Section 8.2). If there is a match, the SRQ line is asserted to notify the controller that an Alarm condition exists.

7.1.2 Least-Significant Nibble

If bit 0 of the Least-Significant Nibble is true (logic 1), a new Peak Value has been received and stored since the last time the Peak Value was read or the Value Status byte received. Likewise, if bit 1 is true, a new Valley Value has been received (see Section 6.7).

If bit 2 is true (logic 1), a Listen Error has been detected. This is a non-fatal error indicating that an unassigned leading character message has been received. It is ignored and the following character, if any, is examined. Therefore, the Listen Error bit is for information only.

7.2 SYSTEM STATUS BYTE

The System Status byte is divided into two nibbles of data and transmitted over the bus in the Nibble format, most-significant nibble first.

	Bit #	System Status Byte	
Most Significant Nibble	7	Send Peak and Valley Values	
	6	Send Average Value	
	5	Send System and Mode Status Bytes	
	4	Send Value Status Bytes	
Least Significant Nibble	3	Spare	
	2	C9-C12	Control I/O Configuration 0=Output 1=Input
	1	C5-C8	
	0	C1-C4	

7.2.1 Most-Significant Nibble

The four bits of the Most-Significant Nibble represent the contents of the data transmitted by the DPM with the Latest Value in the Measurement Message. If the bit is true (logic 1), the corresponding data is included in the Measurement Message. If the bit is false (logic 0), it is excluded.

7.2.2 Least-Significant Nibble

Bits 0,1,2 represent the direction of data flow for the Control I/O lines. The twelve I/O lines are divided into three groups of four lines. All four lines in a group must be the same, either all inputs or all outputs.

If the four lines in a group are chosen to be inputs, the corresponding bit is a logic 1; if outputs, a logic 0.

Bit 2 represents lines C9-C12
Bit 1 represents lines C5-C8
Bit 0 represents lines C1-C4

7.3 MODE STATUS BYTE

The Mode Status byte is divided into two nibbles of data and transmitted over the bus in the Nibble format, most-significant nibble first.

	Bit #	Mode Status Byte
Most Significant Nibble	7	Gated Clock Mode (not used)
	6	Zero Suppression
	5	Talk Only
	4	Compare Average
Least Significant Nibble	3	Line Feed
	2	Carriage Return
	1	Send 1 Reading Only
	0	Triggered Mode

7.3.1 Most-Significant Nibble

If Bit 7 is a logic 1, the Interface is operating in the Gated Clock Mode. This mode is not used with the present DPM's but is included for possible future use. If this mode is accidentally selected by jumper position 8A, the Latest Value will be approximately half that of the DPM display.

If Bit 6 is a logic 1, Zero Suppression has been selected by jumper position 7A and is in effect. See Section 6.2.

If Bit 5 is a logic 1, the Talk-Only mode has been selected by jumper position 6A. However, this cannot be observed because no Status bytes are transmitted in the Talk-Only mode, only the Latest Value. The Interface cannot receive program messages because it cannot be addressed as a listener.

If Bit 4 is a logic 1, the Interface has been programmed with a U1 to compare the Average Value instead of the Latest Value with each of the setpoint values A to D.

7.3.2 Least-Significant Nibble

If Bit 3 is a logic 1, the Interface has been programmed with an O1 to include a line feed with each measurement message unit.

If Bit 2 is a logic 1, the Interface has been programmed with an N1 to include a carriage return with each measurement message unit.

If Bit 1 is a logic 1, the Interface has been programmed with an M1 to send one measurement message only, each time it is addressed as a talker. If Bit 1 is a logic 0, the Interface sends continual readings as long as there is a listener on the bus. When there is no listener, the next value is held in a

buffer and sent the next time there is a listener (see Section 6.5).

If Bit 0 is a logic 1, the Interface has been programmed with an L1 to be in the Triggered mode. A Hold signal is applied to the DPM to prevent it from taking a reading until a GET (Group Execute Trigger) command is received while addressed as a listener or until the Hold signal goes false (high) momentarily.

8.0 SETPOINTS AND ALARMS

8.1 SETPOINTS

Four setpoint values are stored in memory and a comparison made to each one, every time a DPM reading is processed. In the default condition, the Latest Value is compared to the setpoints. If programmed with a U1, the Average Value is compared to the setpoints.

The latter would be recommended for use when the DPM analog input signal has a significant noise component.

Setpoints A to D are initialized with the default value -000000. Other values may be entered with instructions P to S respectively, followed by a value from -999999 to +999999.

Example: Enter the Program Message

P+002000Q+001000R-001000S-002000

for Setpoint A = +2000
Setpoint B = +1000
Setpoint C = -1000
Setpoint D = -2000

Each of these program instructions must contain a header, a sign and 6 digits. Include leading zeros if the values are less than 6 digits. No decimal points are entered in the setpoint values. All comparisons are made without regard for the decimal point position in the DPM reading.

Each time a DPM reading is processed, either the Latest Value (default or program instruction U0) or the Average Value (program instruction U1) is compared to the stored Setpoint values A to D. If the Latest or Average Value is equal to, or more positive than, the setpoint value, the corresponding bit of the Value Status byte is set, otherwise it is reset (see Section 7.1). If the Value Status byte is received, either with every reading (program instruction H1) or on demand (program instruction X9), bits 4 to 7 may be observed for the result of the most recent comparison. The computer controller may take different actions depending on which bits have been set (setpoints reached or exceeded).

If only one pattern of bits 4 to 7 requires action, an Alarm Mask containing that pattern in the Nibble format may be entered with the header V followed by the desired nibble character.

Example:

		D	C	B	A
V9 =	0	0	1	1	1 0 0 1 (see Section 7.0)

In this example, the Alarm Mask pattern 1001 is matched when Setpoint D and A are equalled or exceeded and Setpoints C and B have not been reached. When this condition occurs, an Alarm flag is set (bit 1 of the IEEE Status byte) and the SRQ line is asserted to signal the controller that service is requested. The controller then performs a Serial Poll to identify the device on the bus asserting the SRQ line.

When the IEEE Status byte of the requesting device is read, the SRQ line is unasserted. If the setpoint comparison pattern continues to match the Alarm Mask, the SRQ line again will be asserted next time the DPM reading is requested. To prevent this, the Alarm Mask could be changed to an impossible pattern before the next DPM reading is requested. If the Alarm Mask is set to the next alarm condition pattern, request a DPM reading to enable the DPM to set the SRQ line when the next pattern match occurs.

Use of the Alarm Mask relieves the controller from constantly checking the Value Status byte. The default Alarm Mask pattern is 0110. This prevents a power-on pattern match between the Alarm Mask and the setpoint comparison bits.

9.0 TRIGGERED MODE

The Interface may be placed in the Triggered mode with the program instruction L1. This prevents the DPM from making new measurements until one of the following occurs:

1. Bus trigger by the GET (Group Execute Trigger) command from the controller with the DPM HOLD input high (False).

2. Hardware trigger by a positive (False) pulse on the DPM HOLD input that is normally low true.

When the triggered DPM conversion is complete, the Interface makes a service request by asserting the SRQ line. The controller then conducts a serial poll to determine which device requested service.

When the serial poll is complete, the controller requests the reading from the requesting DPM. The DPM will respond with a reading only to the first request after the serial poll is complete. An additional request will result in a time-out error because the DPM has no new data to send until it is triggered again. It will not repeat a previously transmitted reading (see the note at the end of 10.0 for the only exception).

The GET command is applied to one or more specific devices by making them listeners prior to sending the command.

NOTE: If the SRQ line becomes asserted when going to the Triggered mode with the Program Instruction L1, do an initial serial poll to clear it.

10.0 SERIAL POLL

When the SRQ line is asserted, the controller must poll each device on the bus to determine which device (or devices) is requesting service. It sends the Serial Poll Enable (SPE) command to all devices and then sequentially addresses each device as a talker, one at a time. It listens to the IEEE Status byte sent by the device while it is a talker and if RQS (bit 6) is true, then that device requested service.

After all devices have been polled, the controller knows which one(s) requested service, so it sends the Serial Poll Disable (SPD) command to all devices. It then services the requesting device(s).

The DPM will request service by asserting the SRQ line if:

1. It is in the Triggered mode and a reading, initiated by a software (GET command) or hardware (release of Hold line) trigger, has been completed, or

2. There is a match between the Alarm Mask bit pattern and the setpoint bit pattern in the Value Status byte.

The controller can determine which type of service is requested by analyzing the IEEE Status byte from the DPM read during the serial poll. RQS (bit 6) is true if the DPM is requesting service and Alarm (bit 1) is either false if in the Triggered mode and a new DPM reading is ready, or true if the alarm condition is set by a pattern match between the Alarm Mask and the setpoint bits of the Value Status byte.

Normally, the IEEE Status byte will be sent as a single byte, but it may be repeated in a character string as many times as the controller is willing to listen. When the controller stops listening and asserts the ATN line to send a command or address, the RQS (bit 6) and the Alarm (bit 1) are reset.

CAUTION: Should the DPM be asked to talk again before the Serial Poll Disable (SPD) command is issued, it will again send the IEEE Status byte but with those 2 bits reset.

Whenever serial poll is enabled, the DPM sends the IEEE Status byte; whenever serial poll is disabled, it sends the normal measurement message.

NOTE: During normal operation you cannot read the same Latest value of the DPM more than once. The only exception is when the device has requested service through the SRQ line. This exception is permitted to allow one reading to be obtained after a serial poll, even though the DPM had been read before the serial poll. After the DPM has requested service through the SRQ line, repeated values are read from the DPM until a Serial Poll is executed.

11.0 EXTERNAL CONTROL

The Interface provides 12 digital I/O lines that may be used for external control purposes, completely separate from the DPM functions. The purpose is to allow the user to control unrelated functions from the computer controller without adding a separate IEEE-488 control device to the system. A typical use would be to address an analog multiplexer having multiple input signals and a single output to the DPM.

11.1 CONFIGURING CONTROL I/O LINES

The 12 logic-level I/O lines may be configured as inputs or outputs in groups of 4 lines. The configuration is made with the program instruction T# where # represents a number from 0 to 7. The power-on default value is 7, making all I/O line inputs.

	I/O Lines			
T#	C9-C12	C5-C8	C1-C4	
T0	0	0	0	
T1	0	0	1	
T2	0	1	0	0 = Outputs
T3	0	1	1	1 = Inputs
T4	1	0	0	
T5	1	0	1	
T6	1	1	0	
T7	1	1	1	

Refer to Section 2.0 for the input and output voltage and current specifications.

11.2 READING CONTROL INPUTS

The lines that are configured as inputs are read and put into a buffer at the time the demand program instruction D is received by the Interface. The next time that the DPM is asked to talk, the values in the buffer are sent to the listener(s), typically the computer controller. They are sent as three ASCII characters in the

Nibble format (see Section 7.0). They are sent in the order of the most-significant nibble first and the least-significant nibble last.

Example:

C C C C	C C C C	C C C C
12 11 10 9	8 7 6 5	4 3 2 1
0 0 1 1 1 0 0 1	0 0 1 1 1 1 1 1	0 0 1 1 0 0 0 0
ASCII "9"	ASCII "?"	ASCII "0"

The input data is contained in the four low-order bits of each character. A "1" bit represents a high logic level on the corresponding input line and a "0" bit represents a low level.

All three characters of the input buffer are sent when requested even though some of the 4-line groups may be outputs. Those characters whose four low-order bits represent outputs have input values with no significance and should be ignored.

NOTE: The external inputs are read at the time program instruction D is received and not at the time the values are sent to the controller.

11.3 SENDING CONTROL OUTPUTS

There are three steps in sending data to the control output lines:

1. Use program instruction Z### to put the four low-order bits of each ### character into the output buffer.

2. Use demand program instruction X? to read the output buffer in the Nibble format to ensure that the data was entered properly.

3. Use program instruction F to transfer the data in the output buffer to the latched output lines.

The ### control-output data is transferred on the bus in the Nibble format (see Section 7.0), similar to the control-input data. The ASCII characters are sent in the order of the most-significant nibble first and the least-significant nibble last.

Example:

C C C C	C C C C	C C C C
12 11 10 9	8 7 6 5	4 3 2 1
0 0 1 1 0 1 1 1	0 0 1 1 1 0 1 0	0 0 1 1 0 0 1 0
ASCII "7"	ASCII ":"	ASCII "2"

The output data is contained in the four low-order bits of each character. A "1" bit represents a high logic level on the corresponding output line and "0" bit represents a low level. Therefore, a "0" bit causes the output to sink current and drive a relay, LED, etc.

All three characters must be sent with the program instruction Z###, even though some of the 4-line groups may be inputs. Those characters whose four low-order bits represent inputs are ignored and not transferred to the output lines.

Example:

From the configuration instruction T2,
C9-C12 are outputs
C5-C8 are inputs
C1-C4 are outputs

To sink current with a low level on the eight output lines, send the Program Instruction Z0?0. The "?" character is ignored and could be any of the Nibble format characters.

11.4 CONTROL READY PULSE

Whenever control output data is transferred to the external control lines, a positive 1 usec Control Ready Pulse is output on pin U19.

This occurs about 70 usec after output data transfer to ensure that the lines have settled. New data must have been entered with the program instruction Z### and transferred with the program instruction F before the pulse will appear.

The three step method given at the beginning of this section provides maximum assurance that the outputs are correct when applied to the external control lines. Step 2 may be omitted and steps 1 and 3 combined into one step if this extra security is not warranted.

Example:

Program instruction Z0?0F puts the data in the output buffer and immediately transfers it to the external control lines.

NOTE: Exercise care that a hard positive or negative source, such as an active tri-state output, does not appear on a line that is being used as an output by the Interface.

12.0 CLEAR AND RESET

This section describes the response of the Interface to the IFC, DCL, SDC Controller commands and to the F80A reset program instruction E.

12.1 IFC

The Interface Clear (IFC) command from the controller forces the Interface into its idle state. It removes it from the talker, listener or serial poll enabled states if any are active. It does not clear the talk or listen buffers so talk and listen can be continued later from where they were interrupted.

12.2 DCL, SDC

The Device Clear (DCL) command clears the talk and listen buffers regardless of the current state of the Interface.

The Selected Device Clear (SDC) command clears the talk and listen buffers only if the Interface is an active listener at the time the SDC command is sent.

12.3 POWER-ON RESET

When power is applied to the Interface, the microcomputer starts from a reset program location that initializes memory and sets all program instructions to their default conditions. This effect can be achieved by the Program Instruction E. When this is received by the Interface, an internal flag is set and when the Interface reaches its idle state, the program goes through its power-on reset routine, initializing memory to its default conditions. The talk and listen buffers are cleared.

To accomplish a software power-on reset:

1. Send Program Instruction E to the desired Interface after first addressing it as a listener.
2. Send the IFC command to put the Interface in its idle state. This could also be accomplished with an Unlisten command.

13.0 TALK-ONLY MODE

The Talk-Only mode is selected with jumper position 6A. In this mode, the DPM remains a talker at all times. It may be connected to a device with listen-only capability such as the Newport Model 822 Printer to make a data-logger system that does not require a bus controller.

In this mode, the normal function of the 12 control I/O lines is disabled and they become hard-wired inputs controlling the data output delay time and decimal point position. All program functions assume the power-on default conditions including carriage return and no line feed. The DPM is always a talker and ignores all ATN messages from the controller except SPE and SPD. When ATN or IFC is asserted by the controller, the DPM stops talking and then continues when these are unasserted.

The 12 I/O lines are hard-wired inputs with C1-C7 providing the delay value, C8-C9 the delay units of measurement and C10-C12 the decimal point location.

If the input line is left open, that input is high or a "1".
If the input line is grounded, it is low or a "0".

Delay Value

	C 7	C 6	C 5	C 4	C 3	C 2	C 1	
	X	X	X	X	X	X	X	X = 0 or 1
Weighted value	64	32	16	8	4	2	1	

The delay value is calculated by adding the weighted values of the bits that are "1"'s (open or high).

Delay Units

C 9	C 8	Units of Measurement for the Delay Value
0	0	Seconds
0	1	Minutes
1	0	Hours
1	1	Not used

Decimal Point

DP location # .9.9.9.9.9.9. 7 6 5 4 3 2 1 0 = no decimal point

C 12	C 11	C 10	Decimal Point Locations
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

NOTE: It is suggested that the Control I/O lines C1-C7 be grounded initially in the Talk-Only mode so there is no delay introduced during system setup.

14.0 TROUBLESHOOTING

Refer to the following or other appropriate sections of this manual if difficulty is experienced in achieving proper operation of the F80A option. Section 4 is a particularly good troubleshooting section.

Section 3.2	DPM board jumpers
Section 3.3	DPM and F80A interconnections
Section 3.4	IEEE bus wiring
Section 3.6	Power connections
Section 4.0	Startup
Appendix B	Startup Program

APPENDIX A

Ribbon Cable Assembly

The pin connections for the F80A board accommodate the use of mass-terminated ribbon cable for interconnection with the DPM, the IEEE-488 bus and the control I/O connector. This is an alternate to the PA80 connector assembly option that provides printed circuit interconnections and rigid mounting of the IEEE-488 and control I/O connectors. The ribbon cable assembly is a low cost method of connecting several closely-spaced DPMs to the IEEE-488 bus. It eliminates the expense of standard IEEE-488 cables for this purpose. Most ribbon cable is not shielded and reasonable care must be exercised to keep the lengths short to avoid noise pickup and radiation.

Since the required connector spacing on the ribbon cable is different for each application, this assembly must be fabricated by the user. The following describes the construction details.

Materials

The ribbon cable assembly consists of 50-wire ribbon cable and a number of connectors.

Suggested part numbers are:

Part	Mfg.	Mfg. Part No.
50-wire ribbon cable	3M	3302/50
J1 36-pin PC edge, solder eyelet	SAE	SCU18D/1-2
J2 50-pin PC edge, mass termination	FUJITSU	FCN-767J050-AU/1
J3 24-pin IEEE, mass termination	AMP	552840-1
J4 25-pin D-sub, mass termination	AMP	745242-2

Method

Following are the detailed steps for the construction of a ribbon cable assembly. Refer to Figure A-1.

1. Cut the 50-wire ribbon cable to the total length desired. Attach all 50-pin mass termination connectors J2 at a spacing consistent with the physical mounting of the DPMs.

2. Attach one 24-pin mass termination IEEE-488 connector J3 to the end (or other desired position) of the cable. This is the standard connection to the bus.

3. Cut the 26 lines (wires #25-50) that are not part of the IEEE-488 bus at each connector as shown.

4. Choose a mass-termination connector J4 for I/O Control if this feature is used. It must have a minimum of 16 pins. Starting with wire #25, crimp onto J4 as many wires from wires #25-50 as required to fill the connector and then cut off the outside 10 wires (wires #41-50) of the cable.

5. Peel back the right-hand 8 lines (wires #43-50) to the previous 50-pin connector, separate them, and cut them off at the appropriate length to connect to the lower connector J1 of the DPM.

6. Peel back the adjacent 2 lines (wires #41-42) to the previous 50-pin connector, cut them off as close as possible to the connector, and discard them. These two lines serve as isolation barrier and have no connection on the F80A Interface board.

7. The finished cable connected to the DPMs appears in Figure A-2. The associated wire lists appear in Figures A-3 to A-5.

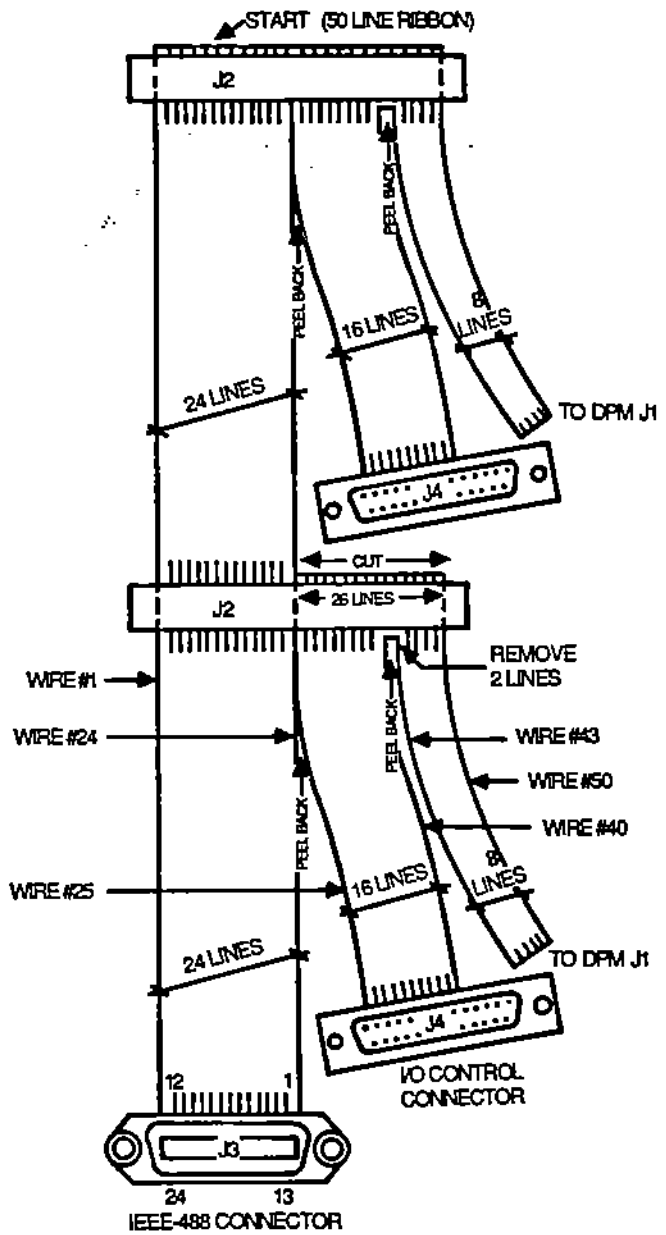


Figure A-1 Ribbon Cable Assembly Construction

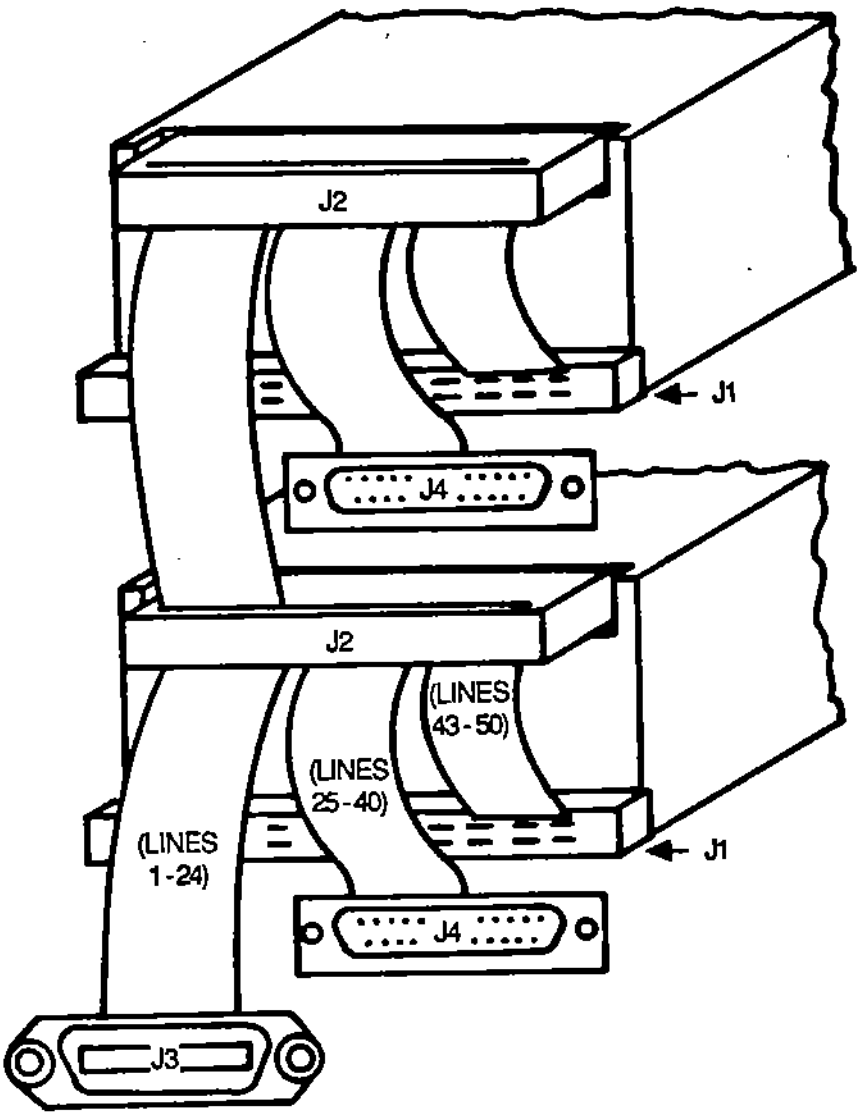


Figure A-2 Ribbon Cable Assembly Installation

F80A UPPER BOARD CONNECTOR	IEEE-488 CONNECTOR	RIBBON CABLE WIRE COLOR	RIBBON CABLE WIRE #	DESCRIPTION
J2-L1	J3-24	Brown	1	Gnd, LOGIC
J2-U1	J3-12	Red	2	SHIELD
J2-L2	J3-23	Orange	3	Gnd, (11)
J2-U2	J3-11	Yellow	4	ATN
J2-L3	J3-22	Green	5	Gnd, (10)
J2-U3	J3-10	Blue	6	SRQ
J2-L4	J3-21	Violet	7	Gnd, (9)
J2-U4	J3-9	Gray	8	IFC
J2-L5	J3-20	White	9	Gnd, (8)
J2-U5	J3-8	Black	10	NDAC
J2-L6	J3-19	Brown	11	Gnd, (7)
J2-U6	J3-7	Red	12	NRFD
J2-L7	J3-18	Orange	13	Gnd, (6)
J2-U7	J3-6	Yellow	14	DAV
J2-L8	J3-17	Green	15	REN
J2-U8	J3-5	Blue	16	EOI
J2-L9	J3-16	Violet	17	DIO8
J2-U9	J3-4	Gray	18	DIO4
J2-L10	J3-15	White	19	DIO7
J2-U10	J3-3	Black	20	DIO3
J2-L11	J3-14	Brown	21	DIO6
J2-U11	J3-2	Red	22	DIO2
J2-L12	J3-13	Orange	23	DIO5
J2-U12	J3-1	Yellow	24	DIO1

Figure A-3 F80A to IEEE-488 Wire List

F80A UPPER BOARD CONNECTOR	I/O CONNECTOR 25-PIN D SUBMINIATURE	RIBBON CABLE WIRE COLOR	RIBBON CABLE WIRE #	DESCRIPTION
J2-L13	J4-13	Green	25	C1
J2-U13	J4-25	Blue	26	C2
J2-L14	J4-12	Violet	27	C3
J2-U14	J4-24	Gray	28	C4
J2-L15	J4-11	White	29	C5
J2-U15	J4-23	Black	30	C6
J2-L16	J4-10	Brown	31	C7
J2-U16	J4-22	Red	32	C8
J2-L17	J4-9	Orange	33	C9
J2-U17	J4-21	Yellow	34	C10
J2-L18	J4-8	Green	35	C11
J2-U18	J4-20	Blue	36	C12
J2-L19	J4-7	Violet	37	GND B
J2-U19	J4-19	Gray	38	Control Ready
J2-L20	J4-6	White	39	GND B
J2-U20	J4-18	Black	40	+5V B

Figure A-4 F80A to Control I/O Wire List

F80 Upper BOARD CONNECTOR	DPM CONNECTOR PIN # 204B	2003B	2004	267B	268	258	RIBBON CABLE WIRE	RIBBON CABLE	SIGNAL NAME
J2-L22	J1-N	J1-N	J1-N	J1-N	J1-N	J1-N	Orange	43	+ POLARITY
J2-U22	J1-R	J1-R	J1-R	J1-R	J1-R	J1-R	Yellow	44	HOLD
J2-L23	J1-P	J1-P	J1-P	J1-P	J1-P	J1-P	Green	45	DATA READY
J2-U23	J1-D	J1-7	J1-7	J1-D	J1-D	J1-D	Blue	46	COUNTER BIT 1
J2-L24	-	-	-	-	-	-	-----	--	----
J2-U24	J1-U	J1-U	J1-U	J1-V	J1-V	J1-S	Gray	48	REFERENCE
J2-L25	J1-T	J1-T	J1-T	J1-T	J1-T	J1-T	White	49	DIGITAL GND
J2-U25	J1-15	J1-15	J1-18	J1-15	J1-15	J1-15	Black	50	+5V OUT

Figure A-5 F80A to DPM Wire List

APPENDIX B

STARTUP Program

B.1 INTRODUCTION

This is an abbreviated program for obtaining a reading from a DPM with an F80A option card. It does not send any programming instructions to the DPM but relies on the default parameters that result from a power-on condition. The program displays either the DPM readings or, after a ten-second timeout, an error code that references an error description in Figure B-4. The purpose is to verify that all cabling and jumpers are correct before proceeding to an application program that sends program instructions to the DPM. Once successful communication between controller and DPM has been established, there is normally no interface problem over the complete range of program instructions and measurement messages. This program assumes that the DPM address is 7, since that is the normal address set at the time of shipment from the factory. If it is set to a different address, it could be changed to 7, or line 150 of the program could be altered to agree with the different address. Section 5.4 of this manual describes how to set the DPM address.

This STARTUP program is written specifically for the National Instruments GPIB-PC Interface board used with the IBM Personal Computer. If a different manufacturer's interface board is used and/or a different computer controller, then the program serves only as a guide. In that case, the program must be rewritten using appropriate nomenclature.

The STARTUP program is written in IBM's BASICA interpretive BASIC language. It may be typed into the computer from the listing that follows or may be obtained, together with the example programs of Appendix C, on the 5 1/4" floppy disk SBØ1 from Newport Electronics, Inc. for a nominal fee.

B.2 NATIONAL INSTRUMENTS INTERFACE

This section applies to the National Instruments GPIB-PC board and software only. It may serve as a guide for other manufacturer's products, but their related manuals must be referenced for detailed instructions and applicable changes to the program.

The National Instruments package consists of their GPIB-PC board that plugs into the IBM PC (or equivalent) and software that interfaces between the board and an application program written in IBM BASIC. This combination allows the IBM PC to serve as a bus controller functioning according to the application program.

Certain preliminary steps must be performed before the program is ready to run.

1. The National Instruments GPIB-PC board must be installed and configured.

2. The software associated with the board must be configured.

3. The pertinent files and programs must be copied to a working disk from which the application program is run.

4. A National Instruments file must be transferred to lines 1 - 6 of the application program.

B.2.1 GPIB-PC Board Installation

From the National Instruments manual, follow the instructions for the hardware configuration and installation of their PC board.

As part of the software configuration, the program IBDIAG is run to confirm that the board is functioning properly.

B.2.2 Software Configuration

Following the National Instruments documentation, copy the files on their distribution disk to a bootable disk by running IBSTART.BAT, which creates or updates CONFIG.SYS to include GPIB.COM with DOS when the system is booted. IBSTART.BAT also runs the hardware diagnostic program IBDIAG, the software configuration program IBCONF, and the software diagnostic program IBTEST.

The software program IBCONF assigns names and configuration parameters to the controller (GPIB-PC) board and devices on the bus. For all of the Newport sample application programs, use the default names of GPIBØ for the board and DEV1-DEV15 for devices.

The set of configuration parameters shown in Figure B-1, entered with the IBCONF program, are known to operate satisfactorily with these application programs. While many of the parameters can be different, this is a set that is known to work and should be tried if there is difficulty in achieving a successful reading.

National Instruments: Board Characteristics	
Board: GPIB0	
Primary GPIB Address	00H
Secondary GPIB Address	NONE
Timeout setting	T10s
EOS byte	00H
Terminate Read on EOS	no
Set EOI with EOS on Write	no
Type of compare on EOS	7-bit
Set EOI w/Last byte of Write	yes
GPIB-PC Model	PC2 or PC2A
Board is System Controller	yes
Local Lockout on all devices	yes
Disable Auto Serial Polling	no
High-speed timing	no
Interrupt jumper setting	NONE
Base I/O Address	2B8H
DMA channel	1

National Instruments: Device Characteristics	
Device: DEV7 Access: GPIB0	
Primary GPIB Address	7
Secondary GPIB Address	NONE
Timeout setting	T10s
EOS byte	00H
Terminate Read on EOS	no
Set EOI with EOS on Write	no
Type of compare on EOS	7-bit
Set EOI w/Last byte of Write	yes

Figure B-1 Configuration Parameters

The program IBCONF saves the configuration parameters in the file GPIB.COM.

B.2.3 Creating the Working Disk

1) Use your DOS system disk to format a new blank working disk using the /S option to put the system file COMMAND.COM on it. Also copy BASICA.EXE to this working disk.

2) Copy all of the programs on the Newport disk SB01 to this working disk and put Newport disk SB01 away for safekeeping. If instead, the program is typed from the STARTUP program listing that follows, save it on this working disk.

3) Copy the following files from the configured National Instrument disk created in B.2.2.

GPIB.COM
BIB.M
CONFIG.SYS

B.2.4 The First Six Lines of the Application Program

The National Instruments file DECL.BAS contains lines 1 - 6 of BASIC code which must be transferred to the first 6 lines of the application programs, STARTUP, FUNCTION and APPLY.

1) Type BASICA and then LOAD "\GPIB-PC\DECL.BAS" from the boot disk.
2) LIST 1-6
3) Insert the working disk from B.2.3 and LOAD "PROGNAME" where PROGNAME is one of the above three program names.
4) Use the screen editor to capture the 6 lines in memory.
5) Substitute the number 55000 (or the result of the calculation in National Instruments documentation) for the XXXXX's in lines 1 and 2.
6) SAVE "PROGNAME"

National Instruments has made several revisions to the software on their distribution disks, including the above file DECL.BAS. Since this file must be compatible with the files GPIB.COM and BIB.M, it is necessary to perform all of the steps in these procedures from the same distribution disk. Do not copy lines 1-6 of the listing in this manual.

The working disk is now ready to use. Type BASICA and then RUN"PROGNAME"; where PROGNAME is one of the application programs, STARTUP, FUNCTION or APPLY.

B.3 EXECUTING THE STARTUP PROGRAM

1. Using a standard IEEE-488 cable, connect the IBM PC or other IEEE-488 controller to the Newport DPM with F80A option and either a PA80 option or custom cable assembly.

2. If the STARTUP program is already on the working disk, jump to item 7, otherwise perform steps 3 - 6 to create the STARTUP program.

3. Type BASICA to get into Basic and then type NEW to clear program space.

4. LOAD "DECL.BAS" from the National Instruments distribution disk. This will load lines 1 - 6 of the STARTUP program. This is the code block that references the calls in BIB.M and it must appear at the beginning of the program. File BIB.M contains the BASICA language interface to the GPIB-PC board.

5. Type the program lines 100 to 470 from the listing in Figure B-2. The explanation of the key program lines is provided for reference in Figure B-3.

6. When done, type SAVE "STARTUP".

7. After connecting the IBM computer controller and the DPM Device to the IEEE-488 bus, insert the working disk. The following programs and files must be on the working disk to successfully run the program STARTUP.

```
COMMAND.COM
BASICA.EXE
STARTUP.BAS
GPIB.COM
CONFIG.SYS
BIB.M
```

Type BASICA to enter BASIC and then type RUN "STARTUP". This program produces a single reading from the DPM for each press of the RETURN key. If there is an error, the error code will be displayed after a ten-second timeout. The error codes are listed in Figure B-4.

Note: The default mode for the DPM and F80A is SEND CONTINUAL. In this mode, the next reading after a transmission is put into the output buffer. Since another reading is not requested until the next key is pressed, this reading will be old data. See Section 6.5 of the manual for further discussion.

```
1       CLEAR   ,55000!          ' BASIC Declarations
2       IBINIT1 = 55000!
3       IBINIT2 = IBINIT1 + 3    ' Lines 1 through 6 MUST be included in your pro
gram.
4       BLOAD "bib.m",IBINIT1
5       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,IBRS
C,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF)
6       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,IBRD,IB
RDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,I
BCNT%)
100 REM   "STARTUP"
110 CLS:WIDTH 80:KEY OFF:COLOR 7,0,0:DEF SEG:POKE 106,0
120 DEFINT A-H,J-Z
130 '**** DEFINE CONSTANTS ****
140 BDNAME$="GPIB0"
150 DVADDRESS=7
160 UNLT$=CHR$(63)+CHR$(95)
170 CL$=CHR$(32)
180 DT$=CHR$(64+DVADDRESS)
190 REM
200 '**** SYS INITIALIZATION ****
210 PRINT "SYSTEM INITIALIZATION":PRINT
220 CALL IBFIND(BDNAME$,BD)
230 GOSUB 440
240 V=0:CALL IBDMA(BD,V)
250 GOSUB 440
260 V=13:CALL IBTMO(BD,V)
270 GOSUB 440
280 CALL IBSIC(BD)
290 GOSUB 440
300 '**** READ DPM ****
310 LOCATE ,,1
320 PRINT "HIT ANY KEY FOR DPM READING, ESC TO END":PRINT
330 KY$=INKEY$:IF KY$="" THEN 330
340 IF KY$=CHR$(27) THEN PRINT "END OF 'STARTUP' PROGRAM":END
350 CMD$=UNLT$+CL$+DT$
360 CALL IBCMD(BD,CMD$)
370 GOSUB 440
380 RD$=SPACE$(70)
390 CALL IBRD(BD,RD$)
400 PRINT:PRINT RD$
410 GOSUB 460
420 GOTO 310
430 '**** ERROR TESTS ****
440 IF IBSTA%<0 THEN PRINT:PRINT "COMMAND ERROR= ";IBERR%
450 RETURN
460 IF IBSTA%<0 THEN PRINT:PRINT "FUNCTION ERROR= ";IBERR%
470 RETURN
```

Figure B-2 STARTUP Program

Lines 1-6	National Instruments code block
Line 110	Clear screen, set width = 80 characters
Line 120	Define variables as integers
Line 140	Assign board name. Should be same as configured in IBCONF.
Line 150	Set DPM address = 7. This should correspond to DPM address jumpers.
Line 160	Unlisten and untalk commands.
Line 170	Controller (IBM computer board) listen address (32+0).
Line 180	DPM talk address (64+7).
Line 220	Returns descriptor BD for board name BDNAME$ "GPIB0".
Line 240	Disables DMA transfers, enables programmed I/O.
Line 260	Sets time limit = 10 sec for I/O operation.
Line 280	Sends 100 microsecond Interface Clear (IFC) signal.
Line 330	Wait for key to be pressed.
Line 340	End program if ESCAPE key is pressed.
Line 350	Create command string of characters for bus.
Line 360	Issue command.
Line 380	Create variable space for receiving reading.
Line 390	Receive reading from DPM.
Line 400	Print reading on screen.
Line 440	If error, print command error number.
Line 460	If error, print function error number.

Figure B-3 Listing Description

Error Code	Explanation
0	DOS error (see IBCNT for DOS error code)
1	Function requires GPIB-PC to be Controller-In-Charge
2	Write function detected no Listeners
3	Interface board not addressed correctly
4	Invalid argument to function call
5	Function requires GPIB-PC to be System Active Controller
6	I/O operation aborted
7	Non-existent interface board
10	I/O operation started before previous operation completed
11	No capability for operation
12	File system operation error
14	Command error during device call
15	Serial Poll status byte lost
16	SRQ remains asserted

Typical Errors	Error Statement
IEEE-488 cable disconnected	Function error = 6
Wrong address (not 7)	Function error = 6
F80A power off	Function error = 6
DPM in HOLD	Function error = 6
Wrong board name (not GPIB0)	Command error = 0
	Function error = 0
Bad command expression (line 350 of program)	Depends on error

Figure B-4 Error Analysis

APPENDIX C

Example Programs

C.1 INTRODUCTION

The two programs listed in this Appendix are written in BASIC and serve as examples for controlling the F80A interface with an IBM PC controller using a National Instruments GPIB-PC interface board. Since the program listings are rather long to type, they may be obtained on a floppy disk from Newport Electronics, Inc. for a nominal fee. Request disk SB01. Also included are compiled BASIC versions of the two programs and the interpretive BASIC program STARTUP, described in Appendix B.

If the National Instruments GPIB-PC board installed in the IBM-PC is to be used as a controller, follow the instructions in Appendix B, Section B.2 for preparation of a working disk from which these programs can be run.

1. "FUNCTION". This menu-driven program contains each of the commands and functions that might be used in application programs. It can be run to demonstrate the IEEE-488 functions and to test the F80A interface.

2. "APPLY". This program contains some of the features that might be included in a typical application program. It is written for 1 or 2 DPMs and an optional printer.

Since these programs are writtenn in IBM Interpretive Basic, they run a little slow. For faster performance, they may be compiled using one of the BASIC compilers for the IBM.

C.2 FUNCTION PROGRAM

The purpose of the test program "FUNCTION" is to:

1. Functionally test the DPM as a stand-alone device and as part of a system.
2. Provide programming examples for each of the IEEE-488 commands.

Type BASICA and then RUN "FUNCTION" to start the program.

Follow the prompts to enter the device addresses in the program. If only 1 DPM is to be exercised by menu selection of function (except for Serial Poll), enter that address and no address prompt will appear after each menu selection. If address prompts are desired for selecting any DPM for any menu selection, enter 99.

In response to the prompt for DPM addresses to be used in the Serial Poll, include the DPM address chosen above to be exercised by menu selection, unless address prompts were chosen (response 99). If there is a printer connected to the bus, enter its address and it can be toggled on or off from the menu. When on, the measurement messages received by the controller will also be received and printed by the printer (e.g., Newport Model 822).

If error messages are displayed when the DPM is exercised by menu selection, refer to the error codes listed in Appendix B, Figure B-4. If the problem persists, re-run the STARTUP program.

The following menu selections are available:

0) **Quit**
End program.

1) **Address Setup**
Change the initial address setup after the program has started the menu prompts.

2) **Single Read DPM**
Obtain one reading from the selected DPM.

3) **Continual Read DPM**
The readings from the selected DPM are continually displayed on a scrolling screen until any key is pushed. The DPM must be in the MØ (continual reading) mode.

4) **Write to DPM**
Allows program instructions to be sent to the selected DPM by prompting for the program message string. Enter all of the desired program instructions in one continuous string. All program instructions are listed on the screen for convenience.

5) **Status Display**
When the Value, System and/or Mode Status values are received from the DPM, they are in the nibble format. To display the description and value of each bit, select STATUS DISPLAY from the menu and enter the nibble format characters. The entry consists of two characters for each value or six characters total in the order of Value, System and Mode. If any are either unknown or of no interest, substitute zeros for them and disregard their section of the display.

6) **IFC**
Sends the Interface Clear command.

7) **DCL**
Sends the Device Clear command to all devices on the bus. This clears the output buffer of the DPMs on the bus.

8) **SDC**
Sends the Selected Device Clear command. This clears only the output buffer of the DPM whose address was either entered during Address Setup or entered from the address prompt following this SDC selection.

9) **GET**
Sends the Group Execute Trigger to all devices that are active listeners on the bus. This program makes all of the DPMs chosen to be included in the Serial Poll, active listeners just prior to sending the Group Execute Trigger signal.

10) **SPE**
Sends the Serial Poll Enable signal. Following this, and until the Serial Poll Disable signal is received, the DPMs send only the IEEE Status byte when addressed to talk.

11) **SPD**
Sends the Serial Poll Disable signal. This returns the DPMs to normal message operation.

12) **Auto Serial Poll**
Uses the National Instruments subroutine IBRSP to perform the Serial Poll on a device level instead of a board level. The SPE and SPD commands are included in the subroutine.

13) **Program Serial Poll**
Sends the SPE command and then reads the IEEE Status byte from each DPM. When complete, it sends the SPD command. The result is the same as the Auto Serial Poll. Either method is acceptable.

14) **Toggle Printer ON/OFF**
This changes the status of the printer as a listener on the bus if it is addressed during the Address Setup routine.

```
1       CLEAR   ,55000!          ' BASIC Declarations
2       IBINIT1 = 55000!
3       IBINIT2 = IBINIT1 + 3    ' Lines 1 through 6 MUST be included in your pro
gram.
4       BLOAD "bib.m",IBINIT1
5       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,IBRS
C,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF)
6       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,IBRD,IB
RDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,I
BCNT%)
90 '
100 ' NAME OF PROGRAM "FUNCTION"
110 '
120 CLS:WIDTH 80:DEF SEG:POKE 106,0:KEY OFF
130 DEFINT A-H,J-Z
140 DIM DPM(15)
150 DIM B1$(7),B2$(7),B3$(7)
190 GOSUB 5210                          'READ DATA STATEMENTS
200 REM **** DEFINE CONSTANTS ****
210 BDNAME$="GPIB0"                     'BOARD NAME
230 UNLT$=CHR$(&H3F)+CHR$(&H5F)         'UNLISTEN, UNTALK COMMAND ADDRESSES
240 SDC=&H4:DCL=&H14                    'SEL DEV CLEAR, DEV CLEAR CMD ADDRESSES
250 SPE=&H18:SPD=&H19                   'SER POLL ENABLE,DISABLE CMD ADDRESSES
260 BGET=&H8                            'GROUP EXECUTE TRIGGER CMD ADDRESS
270 HK$="HIT ANY KEY TO CONTINUE "
280 BK$=SPACE$(79)
290 SRQF=0:SPEF=0                       'SRQ FLAG, SER POLL ENABLE FLAG
300 GOSUB 5810                          'SYSTEM INITIALIZATION
400 GOSUB 5410                          'INITIAL ADDRESS SETUP
490 '
500 '****** MAIN PROGRAM LOOP ******
510 GOSUB 5610                          'MENU
520 INPUT A
530 IF A<0 OR A>14 THEN 520
540 CLS:LOCATE 10,1
550 ON A+1 GOSUB 1010,5410,1210,1610,1810,2010,2210,2410,2610,2810,3010,3210,341
0,3610,3810
560 GOTO 510
990 '
1000 '****** QUIT ******
1010 CLS:LOCATE 5,1
1020 CMD$=UNLT$
1030 GOSUB 4010                         'UNTALK, UNLISTEN
1040 V=0:CALL IBGTS(B0,V)               'ACTIVE CONTROLLER TO STANDBY
1050 GOSUB 4030
1060 COLOR 7:KEY ON:PRINT "END OF PROGRAM"
1070 END
1190 '
1200 '****** SINGLE READ DPM ******
1210 IF D1<>99 THEN 1230
1220 INPUT "DPM # ";D
1230 CMD$=UNLT$+CL$+CHR$(64+D)    'UNLISTEN, UNTALK, CONTROLLER LISTEN, DPM TALK
1240 GOSUB 4010
1250 IF SPEF=0 THEN RD$=SPACE$(70) ELSE RD$=SPACE$(1)
1260 CALL IBRD(B0,RD$)
1270 IF SPEF=1 THEN W=ASC(RD$):GOSUB 4610:GOTO 1310
```

```
1280 PRINT "    ";:LOCATE ,1
1290 L=INSTR(RD$,CHR$(32))
1300 PRINT LEFT$(RD$,L-1)                'PRINT ALL LEFT OF 1ST SPACE
1310 GOSUB 4210
1320 RETURN
1590 '
1600 '****** CONTINUAL READ DPM ******
1610 IF D1<>99 THEN 1630
1620 INPUT "DPM # ";D
1630 CMD$=UNLT$+CL$+CHR$(64+D)   'UNLISTEN, UNTALK, CONTROLLER LISTEN, DPM TALK
1640 GOSUB 4010
1650 KY$=INKEY$:IF KY$<>"" THEN RETURN
1660 GOSUB 1250                          'PART OF SINGLE READ DPM ROUTINE
1670 GOTO 1650
1680 GOTO 1660
1790 '
1800 '****** WRITE TO DPM ******
1810 GOSUB 6000
1820 IF D1<>99 THEN 1840
1830 INPUT "DPM # ";D
1840 INPUT "STRING ";WRT$
1845 CLS:LOCATE 11,1
1850 CMD$=UNLT$+CT$+CHR$(32+D)+CHR$(SDC) 'UNLSN,UNTLK,CONTROLLER TALK,DPM LISTEN
1860 GOSUB 4010
1870 CALL IBWRT(B0,WRT$)
1880 GOSUB 4210
1890 PRINT WRT$+"    "
1900 RETURN
1990 '
2000 '******* STATUS DISPLAY *******
2010 INPUT "STATUS XXXXXX";ST$
2020 FOR J=7 TO 0 STEP -1
2030 K=2-INT(J/4)
2040 W$=MID$(ST$,K,1):GOSUB 5010:A1$=W$
2050 W$=MID$(ST$,K+2,1):GOSUB 5010:A2$=W$
2060 W$=MID$(ST$,K+4,1):GOSUB 5010:A3$=W$
2070 PRINT USING "\              \\\  \              \\\  \              \\\";B1
$(J);A1$;B2$(J);A2$;B3$(J);A3$
2080 NEXT J
2090 RETURN
2190 '
2200 '****** IFC ******
2210 PRINT "WAIT";:LOCATE ,1
2220 CALL IBSIC(B0)
2230 PRINT "IFC "
2240 GOSUB 4030
2250 RETURN
2390 '
2400 '****** DCL ******
2410 CMD$=CHR$(DCL)
2420 GOSUB 4010
2430 PRINT "DCL "
2440 RETURN
2590 '
2600 '****** SDC ******
2610 IF D1<>99 THEN 2630
2620 INPUT "DPM # ";D
2630 CMD$=UNLT$+CHR$(32+D)+CHR$(SDC)
```

```
2640 GOSUB 4010
2650 PRINT "SDC "
2660 RETURN
2790 '
2800 '****** GET ******
2810 CMD$=UNLT$
2820 FOR H=1 TO NUMDEV:CMD$=CMD$+CHR$(32+DPM(H)):NEXT H
2830 CMD$=CMD$+CHR$(SDC)+CHR$(BGET)
2840 GOSUB 4010
2850 PRINT "GET "
2860 RETURN
2990 '
3000 '****** SPE ******
3010 CM$=CHR$(SPE)
3020 GOSUB 4010
3030 PRINT "SPE "
3040 SPEF=1                              'SERIAL POLL ENABLE FLAG
3050 RETURN
3190 '
3200 '****** SPD ******
3210 CMD$=CHR$(SPD)
3220 GOSUB 4010
3230 PRINT "SPD "
3240 SPEF=0
3250 RETURN
3390 '
3400 '******  AUTO SERIAL POLL  ******
3410 V=0:CALL IBONL(B0,V)
3420 FOR H=1 TO NUMDEV
3430 DVNAME$="DEV"+MID$(STR$(DPM(H)),2)
3440 CALL IBFIND (DVNAME$,DV)            'INITIALIZE DEVICE LEVEL
3450 IF DV<0 THEN PRINT "DEVICE FIND ERR"
3460 CALL IBRSP(DV,SPR)
3470 W=SPR
3480 GOSUB 4610                          'ANALYZE IEEE STATUS BYTE
3490 V=0:CALL IBONL(DV,V)                'TAKE DEVICE OFFLINE
3500 GOSUB 4030
3510 NEXT H
3520 GOSUB 5810                         'RE-INITIALIZE BOARD LEVEL
3530 RETURN
3590 '
3600 '******  PROGRAM SERIAL POLL  ******
3610 GOSUB 3010                          'SPE
3620 FOR H=1 TO NUMDEV
3630 CMD$=UNLT$+CL$+CHR$(64+DPM(H))
3640 GOSUB 4010                          'WRITE COMMAND
3650 RD$=SPACE$(1)
3660 CALL IBRD(B0,RD$)
3670 W=ASC(RD$)
3680 GOSUB 4610                          'DISPLAY IEEE STATUS BYTE
3690 GOSUB 4210
3700 NEXT H
3710 CMD$=UNLT$
3720 GOSUB 4010                          'WRITE UNTALK, UNLISTEN
3730 GOSUB 3210                          'SPD
3740 RETURN
3790 '
3800 '******  TOGGLE PRINTER ADDRESS  ******
```

```
3810 IF PA=0 THEN CL$=CHR$(32+BD):PRINT "PRINTER NOT SELECTED":RETURN
3820 L=LEN(CL$)
3830 IF L=1 THEN CL$=CL$+CHR$(32+PA):PRINT "PRINTER ON" ELSE CL$=CHR$(32+BD):PRI
NT "PRINTER OFF"
3840 RETURN
3990 '
4000 '****** IBCMD ROUTINE AND ERROR CHECK ******
4010 PRINT "WAIT";:LOCATE ,1
4020 CALL IBCMD(BO,CMD$)
4030 IF IBSTA%<0 THEN PRINT:PRINT "CMD ERR= ";IBERR%
4040 SRQF=IBSTA% AND &H1000                'SRQ BIT
4050 RETURN
4190 '
4200 '****** FUNCTION ERROR CHECK ******
4210 IF IBSTA%<0 THEN PRINT:PRINT "FNC ERR= ";IBERR%
4220 RETURN
4390 '
4400 '******* STATUS DISPLAY *******
4410 INPUT "STATUS XXXXXX";ST$
4420 FOR J=7 TO 0 STEP -1
4430 K=2-INT(J/4)
4440 W$=MID$(ST$,K,1):GOSUB 5010:A1$=W$
4450 W$=MID$(ST$,K+2,1):GOSUB 5010:A2$=W$
4460 W$=MID$(ST$,K+4,1):GOSUB 5010:A3$=W$
4470 PRINT USING "\               \\\  \              \\\  \              \\\";B1
$(J);A1$;B2$(J);A2$;B3$(J);A3$
4480 NEXT J
4490 RETURN
4590 '
4600 '****** IEEE STATUS BYTE ANALYSIS ******
4610 W$="":G1=0:G6=0
4620 FOR I=7 TO 0 STEP-1
4630 X$="0":IF 2^I<=W THEN X$="1":W=W-2^I
4640 W$=W$+X$
4650 IF I=6 AND X$="1" THEN G6=1
4660 IF I=1 AND X$="1" THEN G1=1
4670 NEXT I
4680 PRINT "DPM #"+STR$(DPM(H))+" IEEE STATUS BYTE = "+W$;
4690 IF G6=1 AND G1=1 THEN PRINT "    *** ALARM ***"
4700 IF G6=1 AND G1=0 THEN PRINT "    *** READ REQUEST ***".
4710 IF G6=0 THEN PRINT
4720 PRINT:RETURN
4790 '
4800 '****** ERASE LINE ******
4810 PRINT SPACE$(79);:LOCATE ,1
4820 RETURN
4990 '
5000 '****** STATUS NIBBLE BIT ANALYSIS ******
5010 IF W$<"0" OR W$>"?" THEN W$=" ":RETURN
5020 N=ASC(W$)-48:G=2^(J-4*(2-K)):W=N AND G:IF W>0 THEN W=1
5030 W$=STR$(W):RETURN
5190 '
5200 '****** STATUS DISPLAY TEXT ******
5210 DATA "NEW PEAK","NEW VALLEY","LISTEN ERROR","SPARE","A","B","C","D"
5220 DATA "1-4","5-8","9-12","SPARE","SEND VAL STS","SEND SYS STS","SEND AVG VAL
","SEND PV VAL"
5230 DATA "REMOTE TRIGGER","SEND 1 ONLY","TERMINATING SEQ","LINE FEED ON","SPARE
","TALK ONLY","ZERO SUPP","GATED CLK MODE"
```

```
5240 FOR J=0 TO 7:READ B1$(J):IF J>3 THEN B1$(J)="OVER SETPOINT "+B1$(J)
5250 NEXT J
5260 FOR J=0 TO 7:READ B2$(J):IF J<3 THEN B2$(J)="CONTROL "+B2$(J)
5270 NEXT J
5280 FOR J=0 TO 7:READ B3$(J):NEXT J
5290 RETURN
5390 '
5400 '******* ADDRESS SETUP *******
5410 CLS:LOCATE 2,20:PRINT "DPM ADDRESS SETUP"
5420 PRINT:PRINT "BOARD (CONTROLLER) CONFIGURED ADDRESS (NORMALLY 0)":INPUT BD
5430 CL$=CHR$(32+BD):CT$=CHR$(64+BD)
5440 PRINT:PRINT "DPM ADDRESS 1-30 FOR MENU RESPONSE (99 FOR MULTIPLE DPMS)"
5450 INPUT D:IF D<1 OR (D>30 AND D<>99) THEN 5440
5460 D1=D
5470 PRINT:PRINT "DPM ADDRESS(ES) 1-30 TO BE USED IN SERIAL POLL (END LIST WITH
0)"
5480 H=1
5490 INPUT DPM(H):IF DPM(H)=0 THEN 5520
5500 H=H+1:IF H<16 THEN 5490
5510 PRINT "ONLY 15 DEVICES ALLOWED"
5520 NUMDEV=H-1
5530 PRINT:PRINT "PRINTER ADDRESS (0 IF NO PRINTER)"
5540 INPUT PA
5550 CLS
5560 RETURN
5590 '
5600 '****** MENU ******
5610 LOCATE 1,1:FOR K=1 TO 9:PRINT BK$:NEXT
5620 LOCATE 1,1,0
5630 PRINT " 0  QUIT                6  IFC (INTERFACE CLEAR)      12  AUTO SERIA
L POLL"
5640 PRINT " 1  ADDRESS SETUP       7  DCL (DEVICE CLEAR)          13  PROGRAM SE
RIAL POLL"
5650 PRINT " 2  SINGLE READ DPM     8  SDC (SELECTED DEV CLEAR)    14  TOGGLE PRI
NTER ON/OFF"
5660 PRINT " 3  CONTINUAL READ DPM  9  GET (GROUP EXECUTE TRIGGER)"
5670 PRINT " 4  WRITE TO DPM       10  SPE (SERIAL POLL ENABLE)"
5680 PRINT " 5  STATUS DISPLAY     11  SPD (SERIAL POLL DISABLE)"
5690 IF SQ>0 THEN LOCATE 14,1:PRINT "*** SRQ ON ***     PERFORM SERIAL POLL (12
OR 13)"
5700 LOCATE 8,1:RETURN
5790 '
5800 '****** SYS INITIALIZATION ******
5810 CALL IBFIND(BDNAME$,B0)
5820 GOSUB 4030
5830 V=0:CALL IBDMA(B0,V)
5840 GOSUB 4030
5850 V=12:CALL IBTMO(B0,V)
5860 GOSUB 4030
5870 CALL IBSIC(B0)
5880 GOSUB 4030
5890 CALL IBSIC(B0):GOSUB 4030          'SEND INTERFACE CLEAR
5900 CMD$=CHR$(DCL):GOSUB 4020          'SEND DEVICE CLEAR
5910 RETURN
6000 '****** PROGRAM INSTRUCTIONS HELP SCREEN ******
6010 LOCATE,,0:CLS:COLOR 15:PRINT "                    PROGRAM INSTRUCTION SUMMAR
Y"
```

```
6020 PRINT "OFF/ON";:COLOR 7:LOCATE ,38,0:PRINT "A  RESET PEAK VALUE"
6030 PRINT " N0/N1 CARRIAGE RETURN                B  RESET VALLEY VALUE"
6040 PRINT " O0/O1 LINE FEED                      C  RESET PEAK & VALLEY VALUES"
6050 PRINT " H0/H1 VALUE STATUS                   D  READ EXT CONTROL INPUTS"
6060 PRINT " I0/I1 SYSTEM + MODE STATUS           E  POWER-ON RESET"
6070 PRINT " J0/J1 AVERAGE VALUE                  F  TRANSFER CNTRL OUTPUT TO POR
T"
6080 PRINT " K0/K1 PEAK + VALLEY VALUES           X0 SEND SETPOINT A VALUE"
6090 PRINT " L0/L1 FREE RUN/TRIGGERED             X1 SEND SETPOINT B VALUE"
6100 PRINT " M0/M1 CONTINUAL/1 READING ONLY       X2 SEND SETPOINT C VALUE"
6110 PRINT " U0/U1 AVG VAL FOR SETPOINT COMPARE   X3 SEND SETPOINT D VALUE"
6120 PRINT " Y#    DEC PT LOCATION. 0=NONE        X4 SEND LATEST VALUE"
6130 PRINT " V#    ALARM MASK: ASCII NIBBLE       X5 SEND AVERAGE VALUE"
6140 PRINT " T#    CONTROL I/O CONFIG: 0-7        X6 SEND PEAK VALUE"
6150 PRINT " Z### CONTROL OUTPUT: ASCII NIBBLE    X7 SEND VALLEY VALUE"
6160 PRINT " P######## SETPOINT A                 X8 SEND ALARM MASK"
6170 PRINT " Q######## SETPOINT B                 X9 SEND VALUE STATUS"
6180 PRINT " R######## SETPOINT C                 X: SEND SYSTEM STATUS"
6190 PRINT " S######## SETPOINT D                 X; SEND MODE STATUS"
6200 PRINT "                                      X< SEND IEEE STATUS"
6210 PRINT "  ASCII NIBBLES=0123456789:;<=>?      X? SEND CONTROL OUTPUT BUFFER"
6220 LOCATE 23,1,0:RETURN
```

C.3 "APPLY" PROGRAM

The purpose of the sample application program "APPLY" is to:

1. Illustrate how a program can be written to perform a sample application task.

2. Provide working application shells of subroutines that can be used directly in custom applications.

C.3.1 Application Model

APPLY is a sample application program using 1 or 2 DPMs and an optional printer that demonstrates the service request mode of operation by changing the alarm mask each reading. There is a match with the setpoint comparisons once every five readings that activates the SRQ line, forcing the IBM controller to perform a serial poll. The screen displays the setpoint values and a bright square when the reading is greater than the setpoint (RDG>). A changing alarm mask is displayed as bright squares indicating the setpoint compare pattern (MASK) that generates a service request. When the two columns of bright squares have the same pattern, the SRQ box displays ON and a serial poll is conducted. The DPM requesting service is identified by the words SERVICE REQUESTED.

C.3.2 Program Model

The APPLY program is structured according to the flow chart of Figure C-1. It contains an IEEE-488 shell of subroutines that is device-independent. The program also contains an application shell that corresponds directly to the F80A interface board. This application shell contains subroutines that call the subroutines of the IEEE-488 shell. Together, these two shells can serve as the root of any customer application program that works with the F80A interface board.

C.3.3 Locating Errors

To facilitate locating the cause of any errors that occur when the APPLY program is run, a variable named ERRLOC is provided that is equated to a location code. This code is printed out when the program terminates on an error and the location can be identified from the error code table, Figure C-2.

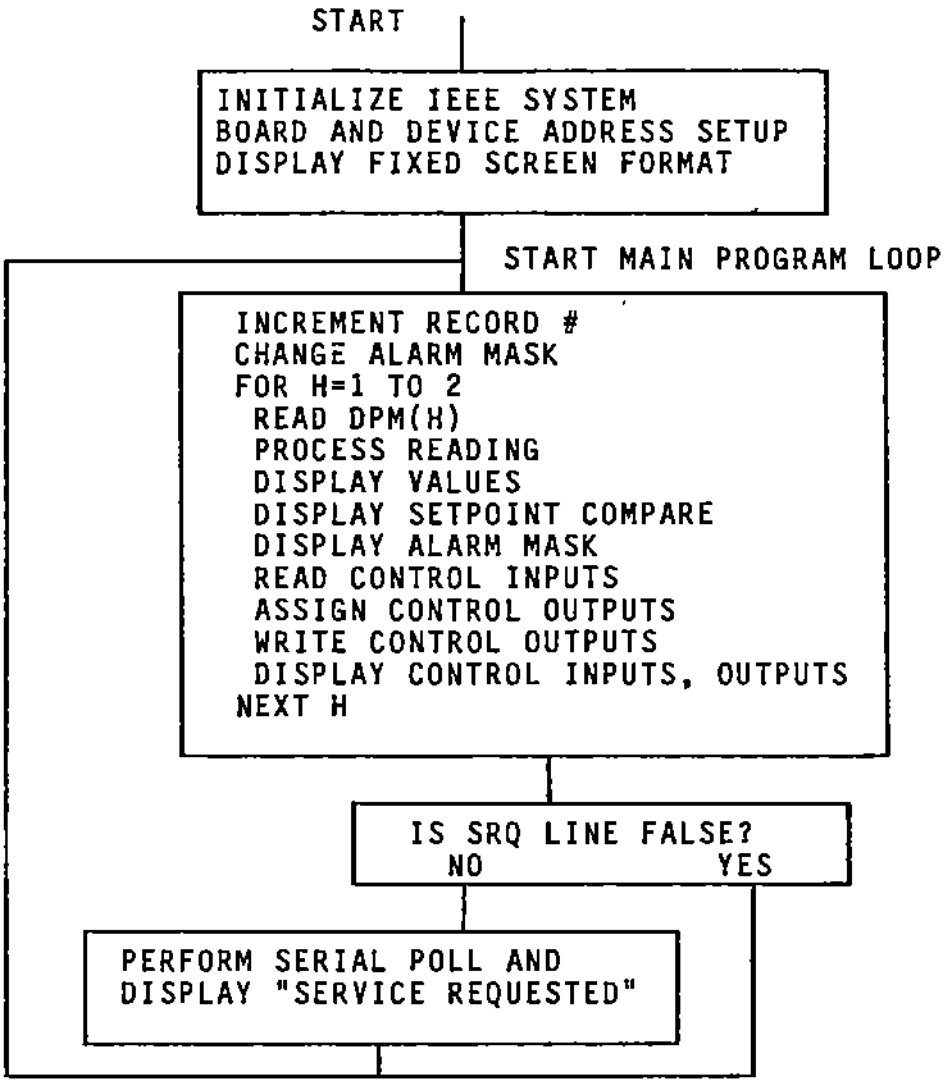


Figure C-1 APPLY Program Flow Chart

ERRLOC CODE	PROGRAM LINE	SUBROUTINE LOCATION	DESCRIPTION
1	5220	1500	IEE Initialization
2	5740	1900	DPM 1 Reset
3	5740	1900	DPM 2 Reset
4	5760	2000	IFC Interface Clear
5	5820	1900	Write Setup data to DPM 1
6	5820	1900	Write Setup data to DPM 2
7	6250	1700	Read data from DPM 1
8	6250	1700	Read data from DPM 2
9	6350	3400	Write control outputs to DPM 1
10	6350	3400	Write control outputs to DPM 2
11	6520	3100	Serial Poll
12	6780	1900	Reset Peak & Valley values

NOTE: If EF = -1 then the CONTROL OUT BUFFER did not agree with the data written to it. This is checked prior to transferring the data to the CONTROL OUTPUT port.

Figure C-2 Error Code Location Table

```
1       CLEAR   ,55000!          ' BASIC Declarations
2       IBINIT1 = 55000!
3       IBINIT2 = IBINIT1 + 3    ' Lines 1 through 6 MUST be included in your pro
gram.
4       BLOAD "bib.m",IBINIT1
5       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,IBRS
C,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF)
6       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,IBRD,IB
RDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,I
BCNT%)
90 '
100 ' NAME OF PROGRAM "APPLY"
110 '
120 'THIS IS A SAMPLE APPLICATION PROGRAM USING 1 OR 2 DPM'S AND AN OPTIONAL
130 'PRINTER THAT PROVIDES BOTH AN IEEE-488 INTERFACE SHELL AND AN APPLICATION
140 'SHELL OF SUBROUTINES FOR USE BY A CUSTOM APPLICATION PROGRAM.
150 '
160 'THIS SAMPLE PROGRAM HAS BEEN DESIGNED TO DEMONSTRATE THE SERVICE REQUEST
170 'MODE OF OPERATION BY CHANGING THE ALARM MASK EACH READING.  THERE IS A
180 'MATCH ONCE EVERY 5 READINGS THAT ACTIVATES THE SRQ LINE, FORCING THE IBM
190 'COMPUTER TO PERFORM A SERIAL POLL.
290 '
300 '****** VARIABLE DEFINITIONS ******
310 ' CL$ = CONTROLLER LISTEN ADDRESS
320 ' CT$ = CONTROLLER TALK ADDRESS
330 ' DL$ = DEVICE LISTEN ADDRESS
340 ' DT$ = DEVICE TALK ADDRESS
350 ' CI(2,12)
360 ' CO(2,12) = CONTROL I/O ARRAYS FOR STORING THE C1 - C12
370 '               INPUTS AND OUTPUTS FOR BOTH DPM'S
380 ' TT(K,J) = TABLE OF BINARY BIT PATTERNS (K=0 TO 3) FOR DECIMAL NIBBLE
390 '             VALUES (J=0 TO 15).  EXAMPLE FOR J=7,
400 '                              TT(3,7)  TT(2,7)  TT(1,7)  TT(0,7)
410 '                                 0        1        1        1
420 ' EF  = ERROR FLAG   1=COMMAND ERROR
430 '                    2=FUNCTION ERROR
440 '                    3=MISC ERROR
450 ' ERRLOC = ERROR LOCATION 1-12
460 ' SRQF = SRQ FLAG    0=OFF  1=ON
490 '
500 '****** DIM STATEMENTS ******
510 DEFINT A-H,J-Q,S-Z:DEFSNG I,R
520 DIM CI(2,12),CO(2,12)                'CONTROL INPUT, OUTPUT ARRAYS
530 DIM TT(3,15)                         'DECIMAL NIBBLE TO BIT TRANSLATION TABLE
540 DIM LV$(2),AV$(2),PV$(2),VV$(2)      'LATEST, AVERAGE, PEAK, VALLEY VALUES
550 DIM A(7),DP(2),DPM(2),W(7),SETPT$(2,4)
560 DIM VALSTS(2,7),SYSSTS(2,7),MODSTS(2,7),IEEESTS(2,7)
590 '
600 '****** CONSTANTS & INITIAL VARIABLE ASSIGNMENTS ******
610 SETUP$="H1J1K1CP+000500Q+001000R+001500S+001900V0T3M1"  'INITIAL DPM SETUP
620             'T3 SETS CONTROL I/O CONFIGURATION FOR
630              '   CONTROL INPUTS = C1-C8   CONTROL OUTPUTS = C9-C12
640 FOR H=1 TO 2:FOR K=1 TO 12
650 CI(H,K)=1:CO(H,K)=1                  'INITIALIZES I/O ARRAYS TO ALL 1'S
660 NEXT K:NEXT H
670 BDNAME$="GPIB0"                     'BOARD ID FROM NATIONAL INST CONFIGURATION
```

```
680 SDC=&H4:DCL=&H14                        'SEL DEV CLEAR, DEV CLEAR CMD ADDRESSES
690 SPE=&H18:SPD=&H19                       'SER POLL ENABLE, DISABLE CMD ADDRESSES
700 BGET=&H8                                'GROUP EXECUTE TRIGGER CMD ADDRESS
710 UNLT$=CHR$(&H3F)+CHR$(&H5F)             'UNLISTEN + UNTALK COMMAND ADDRESSES
720 WW$="0123456789:;<=>?"                  'STRING USED IN ASCII NIBBLE CONVERSION
730 ALRMCNT=0:AC$="0137?"                   'STRING FOR CREATING ALARM MASK NIBBLE
740 V$="0"                                  'INITIAL ALARM MASK
750 RECORD=0                                'RECORD NUMBER
760 '*** CREATE TRANSLATION TABLE ***
770 FOR J=0 TO 15
780 K1=J
790 FOR K=3 TO 0 STEP -1
800 TT(K,J)=INT(K1/2^K):K1=K1-TT(K,J)*2^K   'TABLE OF NIBBLE BIT PATTERNS
810 NEXT K:NEXT J
820 '
830 GOTO 5000                               'START APPLICATION PROGRAM
990 '
1000 '****************************************
1010 '***            IEEE  SHELL             ***
1020 '****************************************
1090 '
1100 '****** IEEE COMMAND ******
1110 CALL IBCMD(B0,CMD$)
1190 '
1200 '****** IEEE COMMAND ERROR CHECK ******
1210 IF IBSTA%<0 THEN EF=EF OR 1            'COMMAND ERROR
1220 IF (IBSTA% AND &H1000)<>0 THEN SRQF=1
1230 RETURN
1290 '
1300 '*** IEEE FUNCTION ERROR CHECK ***
1310 IF IBSTA%<0 THEN EF=EF OR 2            'FUNCTION ERROR
1320 IF (IBSTA% AND &H1000)<>0 THEN SRQF=1
1330 RETURN
1390 '
1400 '****** IEEE MISC ERROR CHECK ******
1410 IF IBSTA%<0 THEN EF=EF OR 4            'MISC ERROR
1420 RETURN
1490 '
1500 '****** IEEE INITIALIZATION ******
1510 EF=0:SRQF=0                            'ERROR FLAG & SRQ FLAG
1520 CALL IBFIND (BDNAME$,B0)               'RETURN UNIT DESCRIPTOR
1530 GOSUB 1410
1540 V=0:CALL IBDMA(B0,V)                   'DISABLE DMA
1550 GOSUB 1410
1560 V=12:CALL IBTMO(B0,V)                  'SET TIMEOUT = 3 SEC
1570 GOSUB 1410
1580 V=1:CALL IBSRE(B0,V)                   'ASSERT REMOTE ENABLE LINE
1590 GOSUB 1410                             'CHECK FOR ERROR
1600 GOSUB 2000                             'SEND INTERFACE CLEAR
1610 GOSUB 2250                             'SEND DEVICE CLEAR
1620 RETURN
1690 '
1700 '****** SINGLE READ DPM ******
1710 DT$=CHR$(64+DPM(H))
1720 CMD$=UNLT$+CL$+DT$
1730 GOSUB 1110                             'SEND COMMAND
1740 RD$=SPACE$(70)                         'SUBROUTINE ENTRY FOR CONTINUOUS READ
1750 CALL IBRD (B0,RD$)                     'READ DATA
```

```
1760 GOSUB 1310                          'CHECK FOR ERROR
1770 RETURN
1890 '
1900 '****** WRITE TO DPM ******
1910 DL$=CHR$(32+DPM(H))
1920 CMD$=UNLT$+CT$+DL$+CHR$(SDC)
1930 GOSUB 1110                          'SEND COMMAND
1940 CALL IBWRT(B0,WRT$)                  'SEND DATA
1950 GOSUB 1310                          'CHECK FOR ERROR
1960 RETURN
1990 '
2000 '****** IFC ******
2010 CALL IBSIC(B0)                      'SEND IFC (INTERFACE CLEAR) COMMAND
2020 GOSUB 1210                          'CHECK FOR ERROR
2030 RETURN
2090 '
2100 '****** SPE ******
2110 CMD$=UNLT$+CL$+CHR$(SPE)
2120 GOSUB 1110                          'SEND SPE (SERIAL POLL ENABLE) COMMAND
2130 RETURN
2190 '
2200 '****** SPD ******
2210 CMD$=UNLT$+CHR$(SPD)
2220 GOSUB 1110                          'SEND SPD (SERIAL POLL DISABLE) COMMAND
2230 RETURN
2240 '
2250 '****** DCL ******
2260 CMD$=CHR$(SDC)
2270 GOSUB 1110                          'SEND DCL (DEVICE CLEAR) COMMAND
2280 RETURN
2290 '
2300 '****** SDC ******
2310 CM$=UQ$+DL$+CHR$(&H4)
2320 GOSUB 1110                          'SEND SDC (SELECTED DEVICE CLEAR) COMMND
2330 RETURN
2390 '
2400 '****** GET ******
2410 DL$="":FOR H=1 TO NUMDEV            'CREATE STRING OF LISTEN ADDRESSES
2420 DL$=DL$+CHR$(32+DPM(H)):NEXT H
2430 CMD$=UNLT$+DL$+CHR$(BGET)
2440 GOSUB 1110                          'SEND GET (GROUP EXEC TRIGGER) COMMAND
2450 RETURN
2490 '
2500 '****** SERIAL POLL ITEM ******
2510 CMD$=CHR$(64+DPM(H))                'H=DPM INDEX
2520 GOSUB 1110                          'MAKE DPM TALKER
2530 RD$=SPACE$(1)
2540 CALL IBRD(B0,RD$)                   'READ IEEE STATUS BYTE
2550 GOSUB 1310                          'CHECK FOR ERROR
2560 W=ASC(RD$):W1=W AND 15:W2=(W-W1)/16 'SEPARATE NIBBLES
2570 W$=CHR$(W2+48)+CHR$(W1+48)          'CONVERT TO ASCII NIBBLES
2580 GOSUB 4010                          'SET IEEE STATUS ARRAY
2590 RETURN
2990 '
3000 '*****************************************************
3010 '***              APPLICATION  SHELL                ****
3020 '*****************************************************
3090 '
```

```
3100 '****** SERIAL POLL ******
3110 GOSUB 2100                          'SPE  SERIAL POLL ENABLE
3120 FOR H=1 TO NUMDEV                   'NUMDEV=NUMBER OF DEVICES
3130 GOSUB 2500                          'SP (SERIAL POLL) ITEM
3140 NEXT H
3150 GOSUB 2200                          'SPD  SERIAL POLL DISABLE
3160 RETURN
3190 '
3200 '****** READ CONTROL INPUTS ******
3210 WRT$="D"                            'D IS COMMAND TO READ CONTROL INPUTS
3220 GOSUB 1900                          'WRITE D
3230 GOSUB 1700                          'READ CONTROL INPUTS
3240 W$=LEFT$(RD$,3)                     'W$ CONTAINS 3 ASCII NIBBLES
3250 FOR J=1 TO 3
3260 W=ASC(MID$(W$,J,1)) AND 15          'W IS 4-BIT NIBBLE
3270 FOR K=3 TO 0 STEP -1
3280 I=13-4*J+K                          'INDEX FOR 12 BIT CONTROL I/O ARRAY
3290 CI(H,I)=TT(K,W)                     'SET CONTROL I/O ARRAY W/ CONTROL INPUTS
3300 NEXT K:NEXT J:RETURN
3390 '
3400 '****** WRITE CONTROL OUTPUTS ******
3410 WRT$="Z"                        'EXTERNAL CONTROL OUTPUT COMMAND
3420 FOR J=1 TO 3                    'CONVERT CONTROL OUTPUT ARRAY TO 3 ASCII NIBBLES
3430 W=0
3440 FOR K=3 TO 0 STEP -1            'CONVERT CONTROL 4-BIT NIBBLE TO ASCII NIBBLE
3450 I=13-4*J+K                      'ARRAY INDEX
3460 IF CO(H,I)=1 THEN W=W+2^K
3470 NEXT K
3480 WRT$=WRT$+MID$(WW$,W+1,1)       'APPEND ASCII NIBBLE
3490 NEXT J
3500 W$=RIGHT$(WRT$,3)               'SAVE CONTROL OUTPUT 3 ASCII NIBBLES
3510 GOSUB 1900                      'WRITE CONTROL OUTPUTS TO DPMS
3520 WRT$="X?"                       'X? INSTRUCTS DPM TO RETURN OUTPUT BUFFER DATA
3530                                 '  FOR VERIFICATION BEFORE TRANSFERRING TO PORT
3540 GOSUB 1900                      'WRITE X? INSTRUCTION
3550 GOSUB 1700                      'READ OUTPUT BUFFER DATA
3560 IF EF<>0 THEN RETURN            'WRITE OR READ ERROR
3570 IF W$<>LEFT$(RD$,3) THEN EF=-1:RETURN  'VERIFICATION ERROR
3580 WRT$="F"                        'NO ERROR. TRANSFER OUTPUT BUFFER TO I/O PORT
3590 GOSUB 1900                      'WRITE "F" INSTRUCTION
3600 RETURN
3690 '
3700 '****** SET VALUE STATUS ARRAY ******
3710 GOSUB 4200                          'CONV 2 ASCII NIBBLES TO VALUE STATUS ARRAY
3720 FOR J=0 TO 7:VALSTS(H,J)=W(J):NEXT:RETURN
3790 '
3800 '****** SET SYSTEM STATUS ARRAY ******
3810 GOSUB 4200                          'CONV 2 ASCII NIBBLES TO SYSTEM STATUS ARRAY
3820 FOR J=0 TO 7:SYSSTS(H,J)=W(J):NEXT J:RETURN
3890 '
3900 '****** SET MODE STATUS ARRAY *)))**
3910 GOSUB 4200                          'CONV 2 ASCII NIBBLES TO MODE STATUS ARRAY
3920 FOR J=0 TO 7:MODSTS(H,J)=W(J):NEXT J:RETURN
3990 '
4000 '****** SET IEEE STATUS ARRAY ******
4010 GOSUB 4200                          'CONV 2 ASCII NIBBLES TO IEEE STATUS ARRAY
4020 FOR J=0 TO 7:IEEESTS(H,J)=W(J):NEXT J:RETURN
4090 '
```

```
4100 '****** CONVERT ASCII NIBBLE TO 4-BIT STATUS ARRAY ******  (NOT USED)
4110            'ENTRY   W$      = 1 ASCII NIBBLE
4120            'EXIT    W(0-3) = EACH ELEMENT 0 OR 1
4130 W=ASC(W$) AND 15
4140 FOR K=0 TO 3                       'CONVERT ASCII NIBBLE TO 4-BIT NIBBLE
4150 W(K)=TT(K,W)                       'TT(K,W) IS NIBBLE TO BIT TRANSLATION
4160 NEXT K:RETURN
4190 '
4200 '*** CONVERT 2 ASCII NIBBLES TO STATUS ARRAY ***
4210            'ENTRY W$ - 2 ASCII NIBBLES
4220            'EXIT W(0-7) - EACH ELEMENT 0 OR 1
4230 FOR J=1 TO 2
4240 W=ASC(MID$(W$,J,1)) AND 15
4250 FOR K=3 TO 0 STEP -1
4260 I=8-4*J+K
4270 W(I)=TT(K,W)
4280 NEXT K:NEXT J:RETURN
4290 '
4300 '****** SEPARATE VALUES ******
4310 J=1                                'VALUES SEPARATED BY CARRIAGE RETURN
4320 IF MID$(RD$,J,1)<>CHR$(13) THEN J=J+1:GOTO 4320
4330 W$=LEFT$(RD$,J-1):RD$=MID$(RD$,J+1)
4340 IF LEN(W$)<8 THEN W$=" "+W$:GOTO 4340  'ADD LEADING SPACES TO MAKE 8 CHARS
4350 RETURN
4390 '
4400 '****** APPLY DECIMAL POINT ******
4410 IF DP=0 THEN W$=" "+W$:RETURN      'APPLY DECIMAL POINT TO VALUES
4420 IF DP=1 THEN W$=W$+CHR$(46):RETURN
4430 W$=LEFT$(W$,8-DP)+CHR$(46)+RIGHT$(W$,DP-1)
4440 RETURN
4490 '
4500 '*** SEPARATE PARAMETERS ***
4510            'ENTRY= RD$,H  WHERE RD$ = LISTEN RESPONSE STRING
4520 W$=LEFT$(RD$,2):RD$=MID$(RD$,4)    'STRIP VALUE STATUS ARRAY
4530 GOSUB 3710                         'SET VALUE STATUS ARRAY
4540 GOSUB 4300                         'STRIP LATEST VALUE
4550 LV$(H)=W$                          'LATEST VALUE
4560 GOSUB 4300                         'STRIP AVERAGE VALUE
4570 AV$(H)=W$                          'AVERAGE VALUE
4580 GOSUB 4300                         'STRIP PEAK VALUE
4590 PV$(H)=W$                          'PEAK VALUE
4600 GOSUB 4300                         'STRIP VALLEY VALUE
4610 VV$(H)=W$                          'VALLEY VALUE
4620 RETURN
4990 '
5000 '*****************************************************************
5010 '***                                                           ***
5020 '***              START APPLICATION  PROGRAM                   ***
5030 '***                                                           ***
5040 '*****************************************************************
5100 '*********************************************
5110 '*              INITIALIZE                    *
5120 '*********************************************
5130 '
5140 CLS:KEY OFF:WIDTH 80:COLOR 7,0,0:DEF SEG:POKE 106,0
5190 '
5200 '****** INITIALIZE IEEE SYSTEM ******
5210 GOSUB 1500                         'IEEE INITIALIZATION
```

```
5220 IF EF<>0 THEN ERRLOC=1:GOTO 7210  'ERRLOC = ERROR LOCATION
5290 '
5300 '****** BOARD & DEVICE ADDRESS SETUP ******
5310 GOTO 5340                          'INTERACTIVE SETUP - DELETE TO BYPASS
5320 BD=0:DPM(1)=7:DPM(2)=8:PA=9        'INTERACTIVE SETUP BYPASS
5330 DP(1)=1:DP(2)=2:NUMDEV=2:GOTO 5560
5340 CLS:PRINT "THIS APPLICATION PROGRAM IS WRITTEN FOR A SYSTEM WITH 1 OR 2 DPM
'S"
5350 PRINT "  AND AN OPTIONAL PRINTER."
5360 PRINT "TYPE 'R' WHILE PROGRAM IS RUNNING TO RESET THE PEAK AND VALLEY VALUE
S."
5370 PRINT "TYPE 'ESC' WHILE PROGRAM IS RUNNING TO END THE PROGRAM"
5380 PRINT
5390 PRINT "ENTER THE IEEE-488 ADDRESSES FOR CONTROLLER, DPM'S AND PRINTER."
5400 PRINT:INPUT "BOARD (CONTROLLER) CONFIGURED ADDRESS (NORMALLY 0)";BD
5410 IF BD<0 OR BD>31 THEN BEEP:GOTO 5400
5420 PRINT:INPUT "NUMBER OF DPM'S  1 OR 2 ";NUMDEV
5430 IF NUMDEV<>1 AND NUMDEV<>2 THEN BEEP:GOTO 5420
5440 PRINT:INPUT "FIRST DPM ADDRESS (1-30)  ";DPM(1)
5450 IF DPM(1)<1 OR DPM(1)>30 OR DPM(1)=BD THEN BEEP:GOTO 5440
5460 INPUT "DECIMAL POINT LOCATION 1-7, 0=NO DP ";DP(1)
5470 IF DP(1)<0 OR DP(1)>7 THEN BEEP:GOTO 5460
5480 IF NUMDEV=1 THEN 5530
5490 PRINT:INPUT "SECOND DPM ADDRESS (1-30) ";DPM(2)
5500 IF DPM(2)<1 OR DPM(2)>30 OR DPM(2)=BD OR DPM(2)=DPM(1) THEN BEEP: GOTO 5490
5510 INPUT "DECIMAL POINT LOCATION 1-7, 0=NO DP ";DP(2)
5520 IF DP(2)<0 OR DP(2)>7 THEN BEEP:GOTO 5510
5530 PRINT:INPUT "PRINTER ADDRESS (0 IF NO PRINTER) ";PA
5540 IF PA=0 THEN 5560
5550 IF PA<0 OR PA>30 OR PA=BD OR PA=DPM(1) OR PA=DPM(2) THEN BEEP:GOTO 5530
5560 CL$=CHR$(32+BD):CT$=CHR$(64+BD)    'CONTROLLER LISTEN & TALK ADDRESSES
5690 '
5700 '****** INITIAL PROGRAM INSTRUCTIONS ******
5710 WRT$="E"                           'RESET COMMAND
5720 FOR H=1 TO NUMDEV
5730 GOSUB 1900                         'RESET DPMS
5740 IF EF<>0 THEN ERRLOC=H+1           'ERRLOC=2,3
5750 NEXT H:IF EF<>0 THEN 7210
5760 GOSUB 2000:IF EF<>0 THEN ERRLOC=4:GOTO 7210  'SEND IFC  ERRLOC=4
5770 FOR J=1 TO 2000:NEXT J              'DELAY FOR POWER-ON RESET TO FINISH
5780           'PROGRAM DPMS TO SEND STATUS BYTE,AVG & PV VALUES, RESET PV-
5790           'SET SETPOINTS, ALARM MASK, I/O CONFIG, 1 READING ONLY
5800 FOR H=1 TO NUMDEV
5810 WRT$=SETUP$+"Y"+RIGHT$(STR$(DP(H)),1)     'ADD DP POSITION TO SETUP
5820 GOSUB 1900:IF EF<>0 THEN ERRLOC=H+4       'ERRLOC=5,6
5830 NEXT H:IF EF<>0 THEN 7210
5890 '
5900 '****** DISPLAY FIXED SCREEN FORMAT ******
5910 GOSUB 8000                         'WRITE FIXED FORMAT TO SCREEN
5920 '
5930 GOSUB 7300                         'WRITE SETPOINTS TO SCREEN
5990 '
6000 '*********************************************
6010 '*        START MAIN PROGRAM LOOP             *
6020 '*********************************************
6100 '****** INCREMENT RECORD, CHANGE ALARM MASK ******
6110 '
6120 SRQF=0:RECORD=RECORD+1             'RESET SRQ FLAG, INCREMENT RECORD #
```

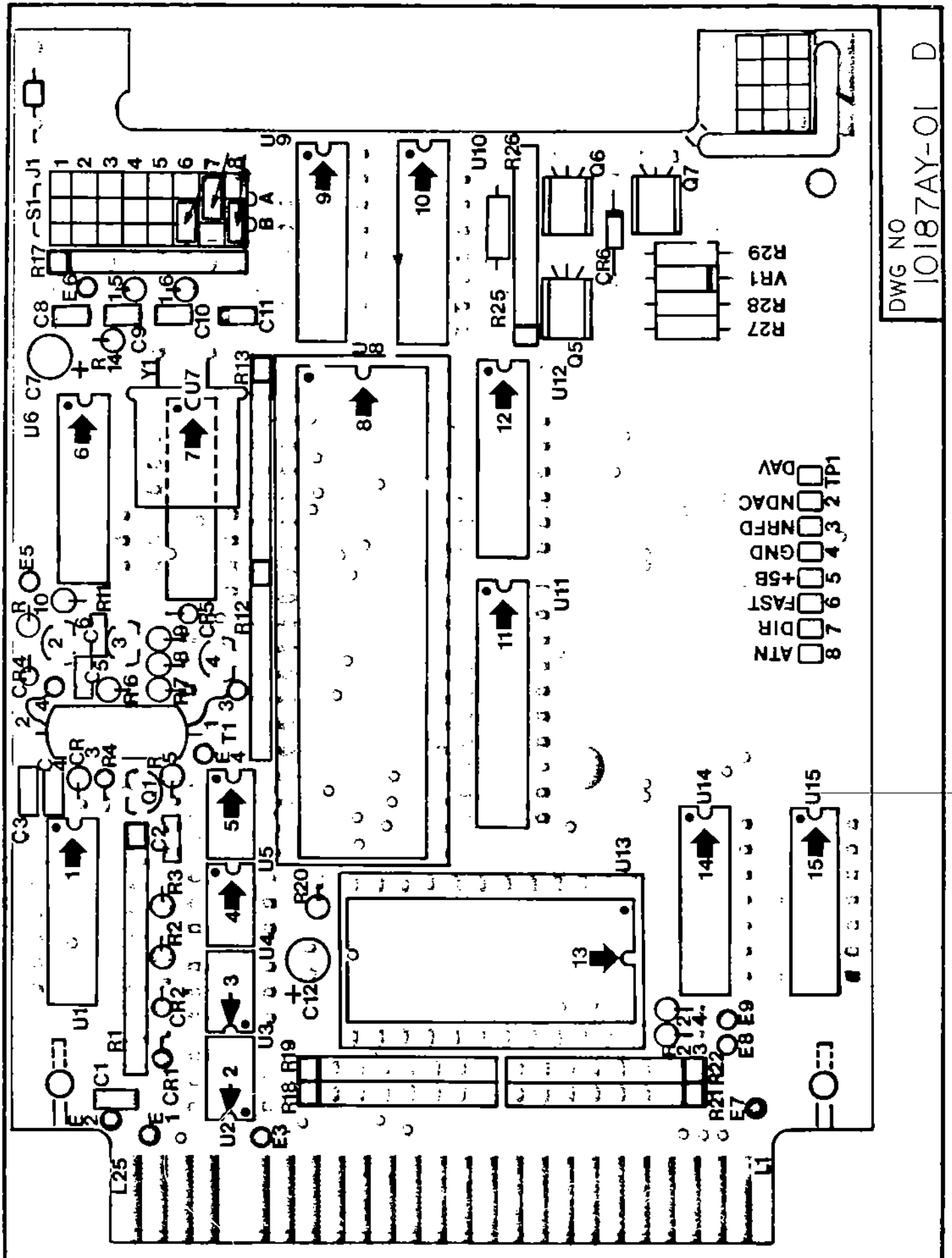
```
6125 IF RECORD>999999! THEN RECORD=1
6130 REC$=MID$(STR$(RECORD),2)          'CREATE 6 CHAR STRING FOR RECORD #
6140 IF LEN(REC$)<6 THEN REC$="0"+REC$:GOTO 6140
6150 COLOR 15:LOCATE 2,3,0:PRINT REC$
6160 GOSUB 7030                         'CHANGE ALARM MASK FOR DEMO PURPOSES
6190 '*******************************************
6200 '*             READ DPMS                   *
6210 '*******************************************
6220 FOR H=1 TO NUMDEV
6230 KY$=INKEY$:IF KY$<>"" THEN GOSUB 6730    'CHECK KEYBOARD
6240 GOSUB 1700                        'READ DATA FROM DPM
6250 IF EF<>0 THEN ERRLOC=H+6:GOTO 7210   'ERRLOC=7,8
6260 GOSUB 4520                        'PROCESS READING
6270 GOSUB 7500                        'DISPLAY VALUES
6280 GOSUB 7600                        'DISPLAY SETPOINT COMPARE (VALUE STATUS)
6290 IF SRQF=1 THEN COLOR 15:LOCATE 23,4:PRINT "ON "
6300 KY$=INKEY$:IF KY$<>"" THEN GOSUB 6730    'CHECK KEYBOARD
6310 GOSUB 7700                        'DISPLAY ALARM MASK
6320 GOSUB 3200                        'READ CONTROL INPUTS
6330 GOSUB 7800                        'ASSIGN CONTROL OUTPUTS FROM PATTERN
6340 GOSUB 3400                        'WRITE CONTROL OUTPUTS
6350 IF EF<>0 THEN ERRLOC=H+8:GOTO 7210  'ERRLOC=9,10
6360 GOSUB 7900                        'DISPLAY CONTROL INPUTS, OUTPUTS TO SCREEN
6390 NEXT H
6400 '*******************************************
6410 '*             CHECK SRQ LINE              *
6420 '*******************************************
6430 LOCATE 22,13,0:PRINT SPC(24)      'CLEAR "SERVICE"
6440 LOCATE 23,13,0:PRINT SPC(24)      'CLEAR "REQUEST"
6450 '
6460 IF SRQF=0 THEN 6680               'SRQ FALSE
6490 '
6500 '****** SERIAL POLL ******
6510 GOSUB 3100                        'PERFORM SERIAL POLL
6520 IF EF<>0 THEN ERRLOC=11:GOTO 7210   'ERRLOC=11
6530 COLOR 15
6540 FOR H=1 TO NUMDEV
6550 IF IEEESTS(H,6)=0 THEN 6620       'CHECK IEEE STATUS BYTE (BIT 6) AND IF
6560 LOCATE 22,14*H-1,0:PRINT "SERVICE"  '  TRUE PRINT "SERVICE REQUESTED"
6570 LOCATE 23,14*H-1,0:PRINT "REQUESTED"
6580 IF PA=0 THEN 6620                 'BYPASS PRINTING ALARM VALUE IF NO PRINTER
6590 DL$=CHR$(32+PA)                   'PRINTER LISTEN ADDRESS
6600 WRT$=" DPM"+STR$(DPM(H))+"  "+LV$(H)+CHR$(13)    'LATEST ALARM READING
6610 GOSUB 1920                        'PRINT ALARM VALUE READING
6620 NEXT H
6630 SRQF=0                            'SRQ FLAG OFF
6640 COLOR 7:LOCATE 23,4,0:PRINT "OFF"
6650 '*******************************************
6660 '*          END MAIN PROGRAM LOOP          *
6670 '*******************************************
6680 GOTO 6120
6690 '
6700 '*******************************************
6710 '*             KEYBOARD CHECK              *
6720 '*******************************************
6730 IF KY$=CHR$(27) THEN 6850
6740 IF KY$<>"R" AND KY$<>"r" THEN RETURN
6750 H1=H:FOR H=1 TO NUMDEV            'REMEM OLD H, THEN RESET PV BOTH DPM'S
6760 WRT$="C"                          'INSTRUCTION TO RESET PEAK & VALLEY VALUES
```

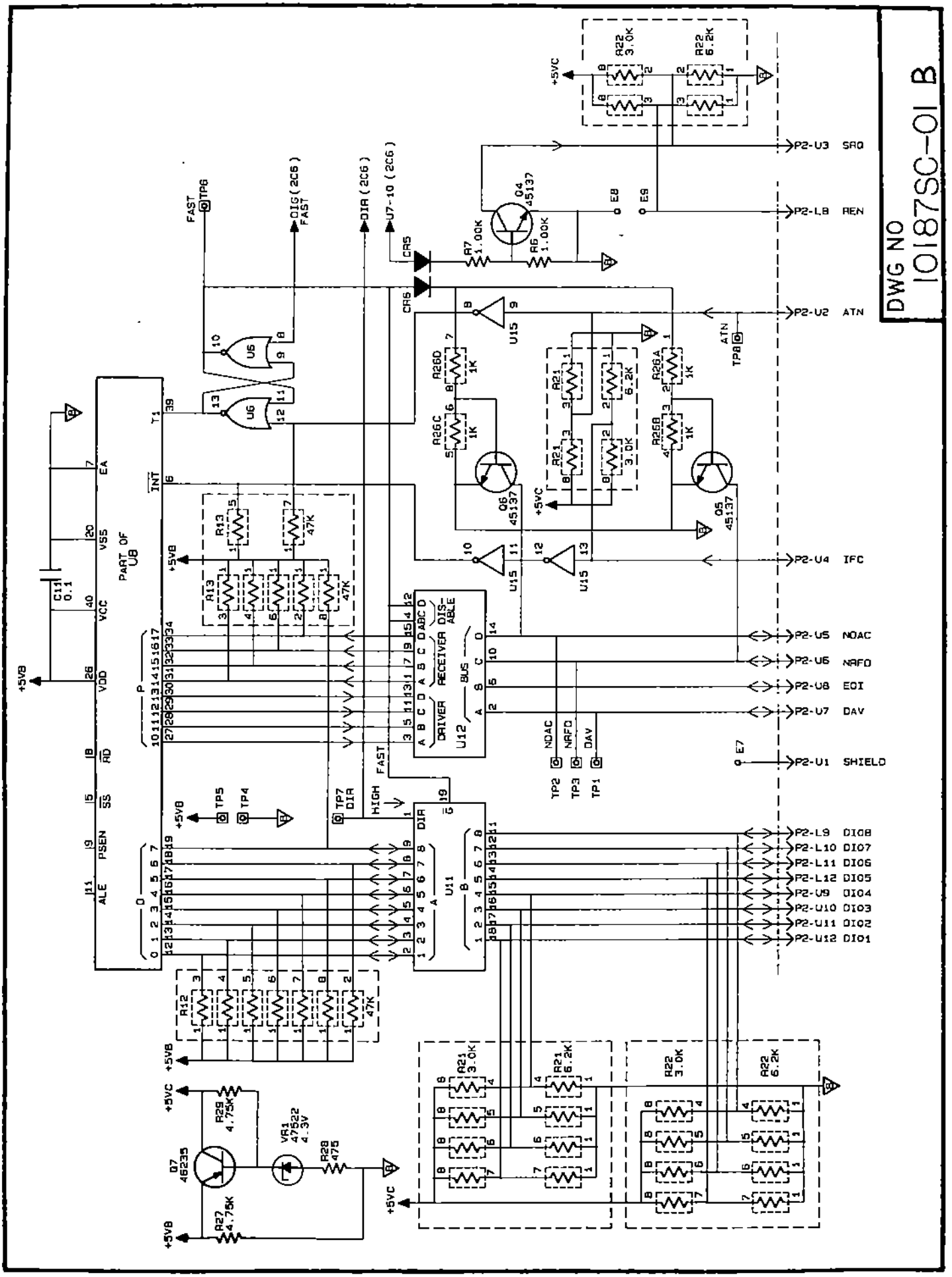
```
6770 GOSUB 1900                          'RESET PEAK & VALLEY VALUES
6780 IF EF<>0 THEN ERRLOC=12:GOTO 7210   'ERRLOC=12
6790 NEXT H
6800 H=H1:RETURN                         'RESTORE OLD H & CONTINUE
6810 '
6820 '*********************************************
6830 '*          END APPLICATION PROGRAM          *
6840 '*********************************************
6850 WIDTH 80:KEY ON:COLOR 7
6860 CMD$=UNLT$
6870 GOSUB 1110                           'UNLISTEN, UNTALK COMMAND
6880 V=0:CALL IBGTS(BO,V)                 'ACTIVE CONTROLLER TO STANDBY
6890 PRINT "END OF 'APPLY' PROGRAM
6900 END
6980 '
6990 '************************************************************
7000 '***       CUSTOM APPLICATION PROGRAM SUBROUTINES         ***
7010 '************************************************************
7020 '
7030 '****** CHANGE ALARM MASK FOR DEMO PURPOSES ******
7040 'INCREMENT ALARM MASK COUNT TO NEXT MASK     ALRMCNT      MASK
7050 '                                              1          0000
7060 '                                              2          0001
7070 '                                              3          0011
7080 '                                              4          0111
7090 '                                              5          1111
7100 ALRMCNT=ALRMCNT+1:IF ALRMCNT=6 THEN ALRMCNT=1
7110 V$=MID$(AC$,ALRMCNT,1)               'GET NEXT NIBBLE IN AC$="0137?" STRING
7120 FOR H=1 TO NUMDEV
7130 DL$=CHR$(32+DPM(H))                  'DPM LISTENER ADDRESS
7140 WRT$="V"+V$
7150 GOSUB 1900                           'WRITE NEW ALARM MASK TO DPM
7160 NEXT H
7170 RETURN
7180 RETURN
7190 '
7200 '****** TERMINATE PROGRAM ON ERROR ******
7210 CLS:KEY ON:WIDTH 80
7220 REM GOTO 15040
7230 PRINT "EF="EF:PRINT "ERRLOC="ERRLOC
7240 PRINT "IBSTA="IBSTA%:PRINT "IBERR="IBERR%
7250 PRINT "END OF 'APPLY' PROGRAM"
7260 END
7290 '
7300 '****** DISPLAY SETPOINTS ******
7310 W$="PQRS"
7320 FOR H=1 TO NUMDEV:FOR J=1 TO 4
7330 L=INSTR(SETUP$,MID$(W$,J,1)):IF L=0 THEN SETPT$(H,J)="":GOTO 7400
7340 SETPT$(H,J)=MID$(SETUP$,L+1,7)
7350 IF DP(H)=0 THEN SETPT$(H,J)=" "+SETPT$(H,J):GOTO 7390
7360             'ADD DP TO SETPOINT
7370 IF DP(H)=1 THEN SETPT$(H,J)=SETPT$(H,J)+CHR$(46):GOTO 7390
7380 SETPT$(H,J)=MID$(SETPT$(H,J),1,8-DP(H))+CHR$(46)+MID$(SETPT$(H,J),9-DP(H))
7390 LOCATE 14-J,14*H-1,0:PRINT SETPT$(H,J)
7400 NEXT J:NEXT H:RETURN
7490 '
7500 '****** DISPLAY VALUES ******
7510 'ENTRY H=1 OR 2
```

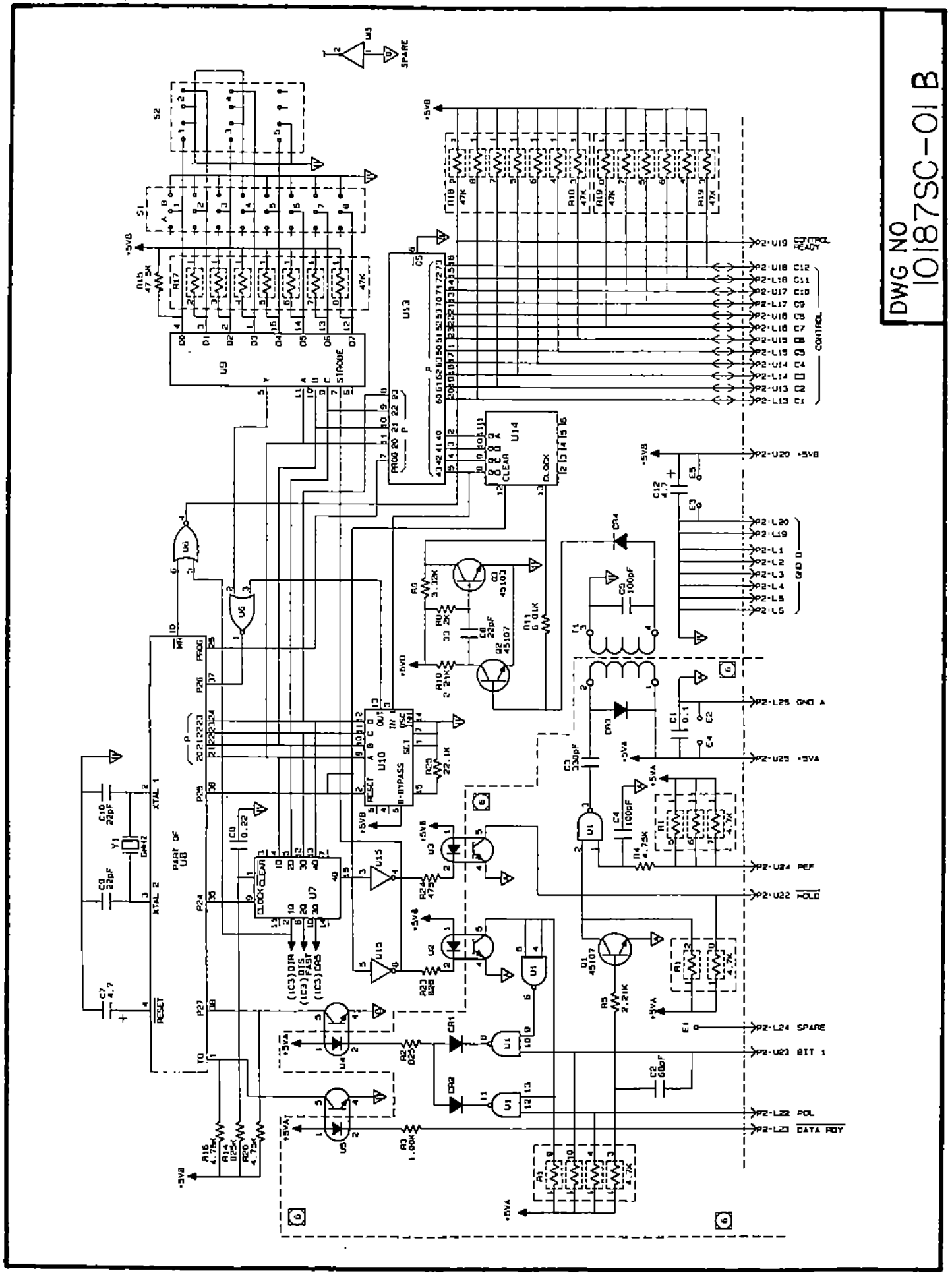
```
7520 COLOR 15
7530 I=H*14-1
7540 LOCATE 5,I,0:PRINT LV$(H)
7550 LOCATE 6,I,0:PRINT AV$(H)
7560 LOCATE 7,I,0:PRINT PV$(H)
7570 LOCATE 8,I,0:PRINT VV$(H)
7580 COLOR 7:RETURN
7590 '
7600 '****** DISPLAY SETPOINT COMPARES ******
7610 COLOR 15:FOR J=4 TO 7
7620 LOCATE 17-J,H*14+8,0:IF VALSTS(H,J)=1 THEN PRINT CHR$(219) ELSE PRINT CHR$(
196)
7630 NEXT J:COLOR 7:RETURN
7690 '
7700  '****** DISPLAY ALARM MASK ******
7710 COLOR 15
7720 W$=V$:GOSUB 4100                    'CONVERT ASCII NIBBLE TO 4-BIT ARRAY
7730 FOR J=0 TO 3
7740 LOCATE 13-J,H*14+10,0:IF W(J)=1 THEN PRINT CHR$(219) ELSE PRINT CHR$(196)
7750 NEXT J:COLOR 7:RETURN
7790 '
7800 '****** ASSIGN CONTROL OUTPUTS FROM PATTERN ******
7810 FOR K=7 TO 4 STEP -1
7820 CO(H,K+5)=VALSTS(H,K)               'CREATE CONTROL OUTPUT ARRAY
7830 NEXT K:RETURN
7890 '
7900 '****** DISPLAY CONTROL INPUTS & OUTPUTS ***
7910 COLOR 15
7920 FOR K=3 TO 0 STEP-1
7930 I=(H-1)*14+20-K
7940 LOCATE 16,I,0:W=CI(H,K+1):PRINT MID$(STR$(W),2)
7950 LOCATE 17,I,0:W=CI(H,K+5):PRINT MID$(STR$(W),2)
7960 LOCATE 19,I,0:W=CO(H,K+9):PRINT MID$(STR$(W),2)
7970 NEXT K:COLOR 7:RETURN
7990 '
8000 '****** FORMAT SCREEN ******
8010 KEY OFF:CLS:WIDTH 40
8020 VL$=CHR$(179):HL$=CHR$(196)         'VERT & HORIZ LINES
8030 VL4$=VL$+SPACE$(4)+VL$
8040 VL8$=VL$+SPACE$(8)+VL$              'VERT LINE + 8 SPACES + VERT LINE
8050 VL10$=VL8$+" "+VL$+" "+VL$
8060 ULC$=CHR$(218):URC$=CHR$(191)
8070 LLC$=CHR$(192):LRC$=CHR$(217)
8080 FOR J=26 TO 12 STEP -14
8090 LOCATE 3,J,0:PRINT TAB(J) " DPM #"STR$(DPM(J/12))
8100 PRINT TAB(J) ULC$+STRING$(8,196)+CHR$(194)+CHR$(196)+CHR$(194)+CHR$(196)+UR
C$
8110 PRINT TAB(J) VL8$+"R"+VL$+"M"+VL$
8120 PRINT TAB(J) VL8$+"D"+VL$+"A"+VL$
8130 PRINT TAB(J) VL8$+"G"+VL$+"S"+VL$
8140 PRINT TAB(J) VL8$+">"+VL$+"K"+VL$
8150 PRINT TAB(J) CHR$(195)+STRING$(8,196)+CHR$(197)+CHR$(196)+CHR$(197)+CHR$(19
6)+CHR$(180)
8160 FOR K=1 TO 4:PRINT TAB(J) VL10$:NEXT
8170 PRINT TAB(J) LLC$+STRING$(8,196)+CHR$(193)+CHR$(196)+CHR$(193)+CHR$(196)+LR
C$
8180 PRINT TAB(J+4) ULC$+STRING$(4,196)+URC$
8190 PRINT TAB(J+4) VL4$
```

```
8200 PRINT TAB(J+4) VL4$
8210 PRINT TAB(J+4)CHR$(195)+STRING$(4,196)+CHR$(180)
8220 PRINT TAB(J+4) VL4$
8230 PRINT TAB(J+4) LLC$+STRING$(4,196)+LRC$
8240 NEXT J
8250 LOCATE 1,1,0
8260 PRINT ULC$+HL$+"RECORD"+HL$+URC$
8270 PRINT VL8$
8280 PRINT LLC$+STRING$(8,196)+LRC$
8290 PRINT:PRINT "LATEST"
8300 PRINT "AVERAGE"
8310 PRINT "PEAK"
8320 PRINT "VALLEY"
8330 PRINT:PRINT "SETPOINT D"
8340 PRINT "SETPOINT C"
8350 PRINT "SETPOINT B"
8360 PRINT "SETPOINT A"
8370 PRINT:PRINT:PRINT "CONTROL  C4-C1"
8380 PRINT " INPUTS  C8-C5"
8390 PRINT:PRINT "CONTROL C12-C9"
8400 PRINT " OUTPUTS"
8410 PRINT
8420 PRINT " "+ULC$+CHR$(196)+"SRQ"+CHR$(196)+URC$
8430 PRINT " "+VL$+" OFF "+VL$
8440 PRINT " "+LLC$+STRING$(5,196)+LRC$;
8450 RETURN
```

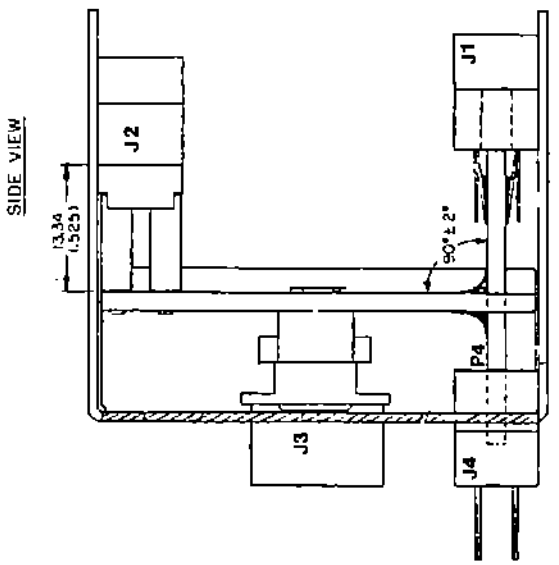
ASSEMBLY - PCBAD, F80A, INTERFACE OPTION

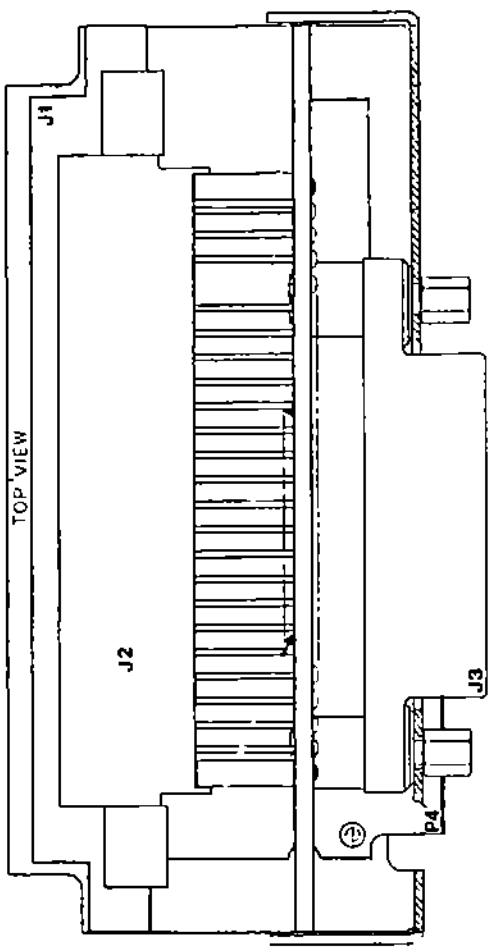


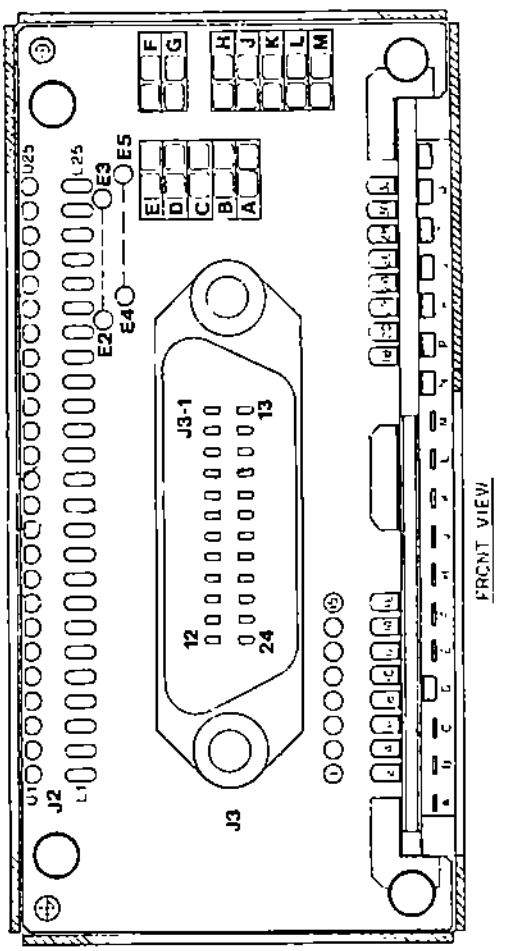

DWG NO
10187SC-01 B
P2-U3 SRQ
P2-L8 REN
P2-U2 ATN
P2-U4 IFC
P2-U5 NDAC
P2-U6 NRFD
P2-U8 EOI
P2-U7 DAV
P2-U1 SHIELD
P2-L9 DIO8
P2-L10 DIO7
P2-L11 DIO6
P2-L12 DIO5
P2-U9 DIO4
P2-U10 DIO3
P2-U11 DIO2
P2-U12 DIO1
FAST
DIG (2C6)
DIA (2C6)
U7-10 (2C6)
PART OF U8
U11
U12
U15
+5VC
+5VB

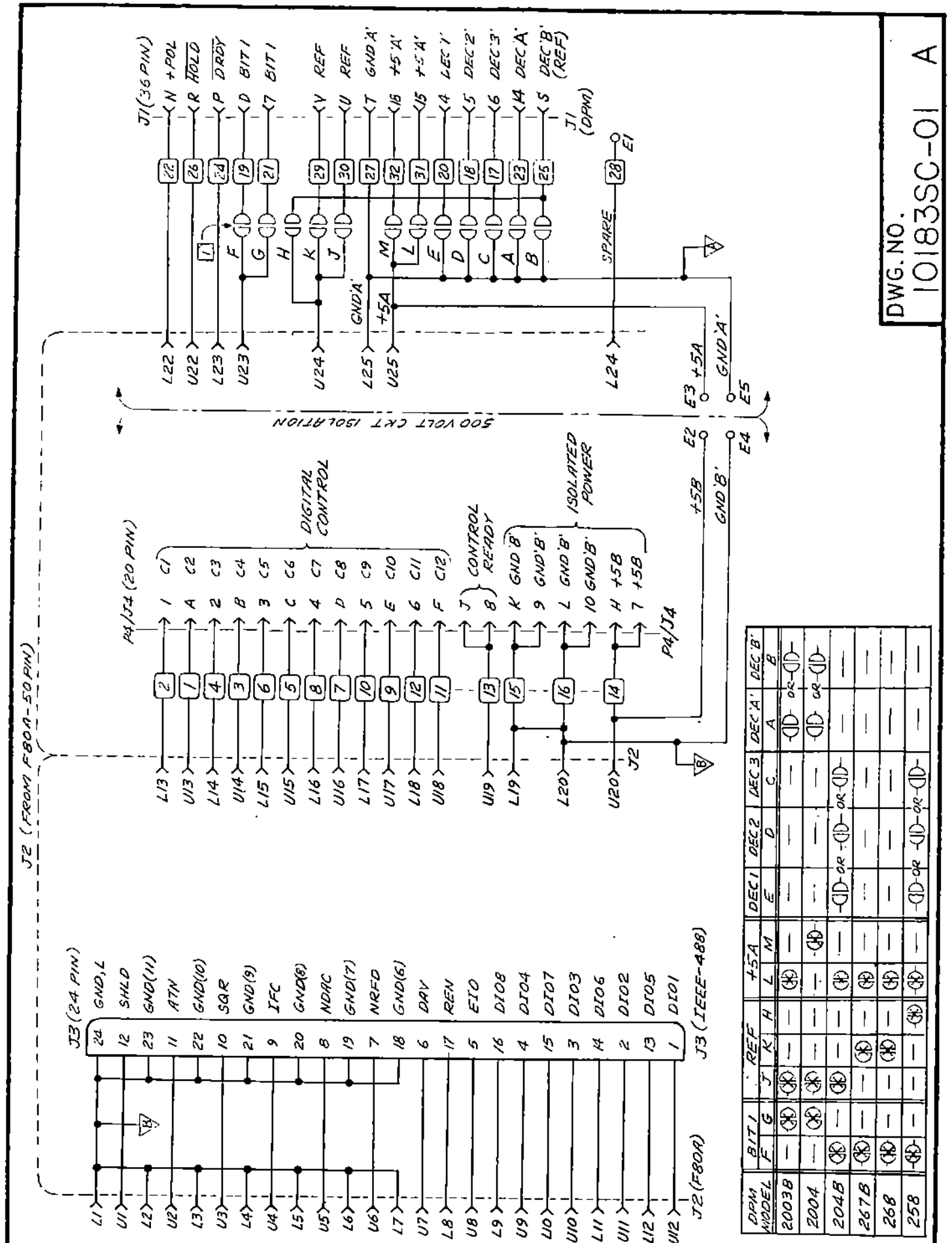

DWG NO
10187SC-01 B

DWG NO
10183AY-01 D








DWG. NO.
10183SC-01 A
J1 (36 PIN)
J1 (DPM)
J2 (FROM F80A-50 PIN)
J3 (24 PIN)
J3 (IEEE-488)
J2 (F80A)
P4/J4 (20 PIN)
P4/J4
J2
500 VOLT CKT ISOLATION
DIGITAL CONTROL
CONTROL READY
ISOLATED POWER
SPARE
+5A
GND'A'
GND'B'
+5B
DPM MODEL
BIT 1
REF
+5A
DEC 1
DEC 2
DEC 3
DEC 'A'
DEC 'B'
2003B
2004
2048
2678
268
258