NATIONAL RADIO ASTRONOMY OBSERVATORY
GREEN BANK, WEST VIRGINIA

ELECTRONICS DIVISION INTERNAL REPORT NO. 246

FRONT-END DATA LINK
USERS' MANUAL

RICHARD J. LACASSE

MARCH 1984

NUMBER OF COPIES: 150

FRONT-END DATA LINK USERS' MANUAL

Richard J. Lacasse

TABLE OF CONTENTS

TABLES

TEST PROCEDURES

FRONT END DATA LINK USERS' MANUAL

Richard J. Lacasse

1.0 Introduction

This manual in intended to supplement Electroncs Division Internal Report No. 239, FRONT-END DATA LINK (FEDAL). It includes sufficient detail for a user to tailor a FEDAL to his application. In particular, the procedure for changing the analog input voltage limits and default initial digital outputs is described in detail. Firmware listings are provided, along with memory maps and a description of the program flow to provide a basis for custom user firmware. Also, test documents for each card and the chassis are included. Finally, FEDAL chassis wiring is specified. The user should become somewhat familiar with EDIR No. 239 before trying to understand the details below.

2.0 System Firmware

This section details the program flow for the V1.0-RJL FEDAL firmware. A memory map is included as an aid to custom programming. Finally, firmware listings are included.

2.1 Program Flow

Four programs control the FEDAL. The first of these, INIT, is normally entered following power-on. It is followed by MAIN, and MAIN may be interrupted by LINT or UACINT.

The INIT program initializes FEDAL analog outputs to 0 V and digital outputs to default values specified in ROM. It then tests the RAM, ROM, and ACIA's, setting status bits, if appropriate. Finally, it initializes the RAM and ACIA's, enables interrupts, and jumps to MAIN.

MAIN simply calls a number of subroutines to accomplish the following tasks: pulsing the auto-reset circuit and monitoring the link data lines (LPMON), acquiring, testing, and transmitting data (SYNCTR, ANACTR, DIGINT, ENDTR), and displaying received data (DISPLY). The MAIN program loops indefinitely until interrupted.

The LINT program is entered whenever the link ACIA interrupts the microprocessor to signal that a data byte has been received. LINT accepts and, if necessary, stores the byte. It keeps track of where in the data frame it is, and calculates a check-sum to compare with the transmitted one. Upon detection of a complete, good data frame it updates the dedicated user outputs.

The UACINT program is entered by interrupt whenever the user ACIA receives a byte. Since no user interface is defined, the program simply resets the interrupt request by reading from the ACIA. This program is included to provide a "hook" for a custom user program.

3.0 Memory Map

The FEDAL address space is divided as shown in Table 1. The reason certain devices, like the LINK ACIA, occupy a large address space is that only the five most significant address bits are decoded to select these devices. Detailed RAM and ROM maps are shown in Tables 2 and 3, respectively.

TABLE 1

Front-End Data Link Address Space

Device	Address (Hex)	Description	
RAM	0000 to 03FF		
AUTORESET	10XX		
LINKACIA	18XX	00 - Control & Status Register 01 - R/W Data Buf	
OISWITCH	20XX	Link Input MUX Control	
UACIA	28XX	00 -- Control & Status Register 01 -- R/W Data Buf	
ANALOG INPUT BOARD	48XX	00 -- MUX Latch 01 -- Start Convert 02 -- Status & 4 MSB's (+ 2 Monitors) 03 -- 8 LSB's	
ANALOG OUTPUT SLOT 3	5000 :		MSBØ
	5001 :-	Analog Out 0 & 1	LSN-Ø, MSN-1
	5002 :		LSB1
	5003 :		
	5004 :-	Analog Out 2 & 3	
	5005 :		
	5006 :		
	5007 :-	Analog Out 4 & 5	
	5008 :		
	5009 :		
	500A :-	Analog Out 6 & 7	
	500B :		
	500C :		
	500D :-	Analog Out 8 & 9	
	500E :		
	500F :		
	5010 :-	Analog Out 10 & 11	
	5011 :		
	5012 :-	Digital Outputs 0 to 7	
	5013 :-	Digital Outputs 8 to 15	

Table 1 - Page 2

Device	Address (Hex)	Description
ANALOG OUTPUT SLOT 4	5800 :	
	01 :	Analog Output 12 & 13
	02 :	
	03 :	
	04 :	Analog Output 14 & 15
	05 :	
	06 :	
	07 :	Analog Output 16 & 17
	08 :	
	09 :	
	0A :	Analog Output 18 & 19
	0B :	
	0C :	
	0D :	Analog Output 20 & 21
	0E :	
	0F :	
	10 :	Analog Output 22 & 23
	11 :	
	12 :	Digital Output 16 to 23
	13 :	Digital Output 24 to 31
ANALOG OUTPUT SLOT 5	6000 :	
	01 :	Analog Output 24 & 25
	02 :	
	03 :	
	04 :	Analog Output 26 & 27
	05 :	
	06 :	
	07 :	Analog Output 28 & 29
	08 :	
	09 :	
	0A :	Analog Output 30 & 31
	0B :	
	0C :	Analog Output Monitor
	0D :	0 & 1 (Front Panel)
	0E :	
	0F :	Analog Output Monitor
	10 :	2 & 3 (Rear Panel)
	11 :	
	12 :	Internal Control Bits
	13 :	and Spares

Table 1 - Page 3

Device		Address (Hex)	Description
LOCAL DISPLAY (FRONT PANEL)		68XX	Local Display Latch
REMOTE DISPLAY (FRONT PANEL)		70XX	Remote Display Latch
DIGISWITCH	0	80XX	Controls AN.MON 0, Front Panel DVM
	1	88XX	Controls AN.MON 1, Front Panel AVM
	2	90XX	Controls AN.MON 2, User Monitor
	3	98XX	Controls AN.MON 3, User Monitor
	4	A0XX	MSB used to select limit or Digital Outputs
	5	A8XX	Spare
TOGGLE SWITCH	0	B000	Digital Inputs 0 - 7
	1	B001	Digital Inputs 8 - 15
	2	B002	Digital Inputs 16 - 23
	3	B003	Digital Inputs 24 - 31
ROM		F000 to FFFF	

TABLE 2

Memory (RAM) Detail
(Addr 0000 to 03FF)

Address (Hex)		Symbol	Description
0000		Spare	
.		.	
.		.	
.		.	
000F		Spare	
0010	NIRTBL	CH 0-7 |	One bit for each analog
0011		CH 8-15 |	input channel.
0012		CH 16-23 |	A "1" means not in range
0013		CH 24-31 |	specified.
0014		SPARE	
0015		MUXCT	Counter for Analog MUX.
0016		SPARE	
0017		STATUS	Local Status Word.
0018		SPARE	
0019		SPARE	
001A		SPARE	
001B		SYMOCT	LINT Sync Mode (Init. to 2).
001C		SPARE	
001D		HDRCT	LINT Header Byte Count
001E		SPARE	
001F		BYTECT	LINT Data Byte Count.
0020		RBUFØ	A buffer for received data.
0021		.	(Approx. 16 spare locations.)
.		.	
.		.	
006F		RBUFØ	
0070		RBUF1	A second buffer for received data.
.		.	
.		.	
.		.	
00BF		RBUF1	
00C0		SPARE	
00C1		RBUFSE	Flag to select one of above buffers.
00C2		SPARE	
00C3		CRCCOM	Computed CRC.
00C4		SPARE	
00C5		BFRACT	Bad frame counter.
00C6		SPARE	
00C7		FRRDY	"FRAME READY" flag.
00C8		SPARE	
00C9		REMSTA	Remote Status.
00CA		SPARE	
00CB		FSYNOK	First Sync OK.
00CC		SPARE	
00CD		BSYNCT	Bad sync word ctr.
00CE		SPARE	

Table 2 - Page 2

Address (Hex)		
00CF	ATEMP1	Used for temporary storage of
00D0	TIPLUS	miscellaneous quantities.
00D1	CRCLOC	Transmitted CRC, generated by sending programs.
00D2	SPARE	
00D3	LCOMER	Locally detected communication errors.
00D4	SPARE	
00D5	BLKDIS	Display blanking byte.
00D6	SPARE	
00D7	RCVDGD	Flag first good data frame.
00D8	SPARE	
00D9	FROUT	Counter reset when Data -> user incremented once per program loop.
00DA	SPARE	
00DB	CHANI	Memory image of selected link channel.
00DC	SPARE	
00DD	CKNEXT	Flag to retest other link chain.
	SPARE	
00DF	RCVPTR	Used to point to next addr when storing received data.
	SPARE	
00E1	CH0GD	Channel 0 good flag.
00E2	CH0TST	Channel 0 tested flag.
00E3	CH1GD	Channel 1 good flag.
00E4	CH1TST	Channel 1 tested flag.
00E5	SPARE	
.	.	
.	.	
.	.	
02FF	SPARE	
0300	STACK	
.	.	
.	.	
.	.	
03FF	STACK	

TABLE 3

ROM Memory Map

Address From	Address To	Routine or Data
F000	F131	INIT
F140	F153	MAIN
F170	F18A	DSWRD
F300	F303	UACINT
F500	F5AC	ANACTR
F7FD		CHECK SUM ADJUST WORD FOR 6A ROM.
F7FE	F7FF	6A ROM CHECK SUM.
F800	F87F	ANALOG LIMIT TABLE.
F880		CHECK SUM ADJUST WORD FOR 10A ROM.
F881	F882	10A ROM CHECK SUM.
F888	F88D	INITIAL DIGITAL OUTPUTS.
F890	F8CD	LIMCK
F8E0	F906	CONVRT
F910	FA39	LINT
FAE0	FAF8	LIMSET
FB00	FB1E	LXMIT
FB30	FB38	SYNCTR
FB40	FB53	DIGNIT
FB58	FB7F	ENDTR
FB88	FBCA	GET2
FCE0	FD99	DISPLY
FDE0	FDE7	AOUT
FDF0	FE24	UPDATE
FE60	FF16	LPMON
FFF0	FFFF	INTERRUPT VECTORS

4.0 Data Communication Format

Table 4 depicts the format of the data stream on the link. The number of data bytes in the frame can be changed; however, the byte count (Byte #3) should be changed to reflect this. The LINT subroutine uses the byte count in accepting data, and so is flexible; however, certain programs that pick data out of the received frame and others which generate the frame are not so general. Care must be taken if changing the number or significance of the data bytes to change all affected subroutines.

TABLE 4

Data Communication Format

Byte #	Contents (Hex)	Description
0	7E	1st SYNC WORD
1	7E	2nd SYNC WORD
2	XX	STATUS
3	3B	# OF BYTES/FRAME (HEX) OF DATA
4	XX	ANOUT 0- 1 (TRANSMITTED ANALOG VOLTAGES)
5	XX	
6	XX	
7	.	ANOUT 2- 3
8	.	
9	.	
A		ANOUT 4- 5
B		
C		
D		ANOUT 6- 7
E		
F		
10		ANOUT 8- 9
1		
2		
3		ANOUT 10-11

Table 4 - Page 2

Byte #	Contents (Hex)	Description
14		
15		
16		ANOUT 12-13
17		
18		
19		ANOUT 14-15
1A		
1B		
1C		ANOUT 16-17
1D		
1E		
1F		ANOUT 18-19
20		
21		
22		ANOUT 20-21
23		
24		
25		ANOUT 22-23
26		
27		
28		ANOUT 24-25
29		
2A		
2B		ANOUT 26-27
1C		
2D		
2E	.	ANOUT 28-29
2F	.	
30	.	
31	XX	ANOUT 30-31
32	XX	
33	XX	
34	XX	DIGOUT MSB (TRANSMITTED DIGITAL INFO)
35	XX	DIGOUT
36	XX	DIGOUT
37	XX	DIGOUT LSB
38	XX	LIM 0- 7
39	XX	LIM 8-15
3A	XX	LIM 16-23
3B	XX	LIM 24-31
3C	XX	COMERR
3D	XX	RESTERR
3E	XX	CRC CHECK (Sum, mod 256, bytes 2 through 3D).

Table 4 - Page 3

Notes:

1. Each byte: 1 START BIT
   8 DATA BITS
   1 STOP BIT

   10 BITS TOTAL

   BIT TIME = 16 us

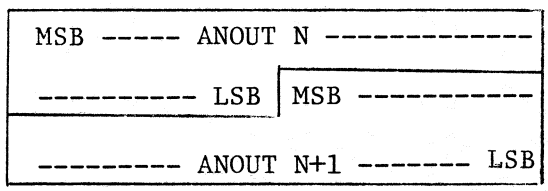
   "BYTE" TIME = 160 us

   FRAME TIME = 9.920 ms

   (This max data rate will not be met due to time consuming interrupt and acquisition programming.)

2. Two analog voltages, 12 bits each, are formatted into three bytes as follows:

   Byte #



5.0 Firmware Listings

This section contains listings of all the routines in the V1.0-RJL FEDAL system. Each listing contains a header with a program description, a symbol definition table, and the code with embedded comments. Thus, it is self-documenting. The code is written in American Automation's version of Motorola 6809 Assembly Language.

LISTINGS FOLLOW.

```
AMERICAN AUTOMATION ASSEMBLER          ANACTR  20  Dec  83          PAGE 001

0001                        ********************************************************
0002                        *                                                      *
0003                        * ANACTR SUBROUTINE                                    *
0004                        *                                                      *
0005                        *   FUNCTION: TRANSMITS STATUS, NUMBER OF BYTES,AND    *
0006                        *             32 ANALOG CHANNELS. CONTROLS THE ACQUI-  *
0007                        *             SITION OF ANALOG INPUTS                  *
0008                        *   INPUT:    STATUS WORD                              *
0009                        *   OUTPUT:   STATUS,NUMBER OF BYTES, AND 32 ANALOG    *
0010                        *             CHANNELS TO LINK UART.                   *
0011                        *   CALLS:    LXMIT, CONVRT, LIMCK, LIMSET             *
0012                        *   USES:     D,U,X,Y,S-STACK                          *
0013                        *                                                      *
0014                        ********************************************************
0015                        *
0016                        *
0017                        ***************SYMBOL DEFINITIONS***********************
0018                        *
0019                        *LIMCK IS A SUBROUTINE WHICH CHECKS THE LIMITS OF THE
0020                        *   ANALOG INPUTS.
0021  A 0000 0000F890        LIMCK    EQU     $F890
0022                        *
0023                        *LIMSET IS A SUBROUTINE WHICH SETS A LIMIT BIT
0024  A 0000 0000FAE0       LIMSET   EQU     $FAE0
0025                        *
0026                        *LXMIT IS A SUBROUTINE WHICH LOADS DATA TO THE LINK
0027                        *   TRANSMIT BUF FROM REG. A.
0028  A 0000 0000FB00       LXMIT    EQU     $FB00
0029                        *
0030                        *CONVRT IS A SUBROUTINE WHICH INITIATES AN A/D CONVER-
0031                        *   SION AND WAITS FOR A READY STATUS. A TIME OUT BY
0032                        *   CONVRT IS INDICATED TO ANACTR BY CARRY=1.
0033  A 0000 0000F8E0       CONVRT   EQU     $F8E0
0034                        *
0035                        *TEMP1 IS A TEMPORARY STORAGE LOCATION
0036  A 0000 000000CF       TEMP1    EQU     $CF
0037                        *
0038                        *MUXCT IS USED TO KEEP UP WITH THE CHANNEL TO CONVERT
0039  A 0000 00000015       MUXCT    EQU     $15
0040                        *
0041                        *MUXLT IS THE ADDRESS OF THE MUX LATCH
```

```
AMERICAN AUTOMATION ASSEMBLER          ANACTR  20  Dec  83          PAGE 002

0042   A 0000 00004800      MUXLT    EQU     $4800
0043                        *
0044                        *NUMBYT IS THE NUMBER OF BYTES TRANSMITTED IN THE FRAME
0045   A 0000 0000003B      NUMBYT   EQU     $3B
0046                        *
0047                        *STATUS IS THE LOCAL STATUS WORD
0048   A 0000 00000017      STATUS   EQU     $17
0049                        *
0050                        *CRCLOC IS THE CRC GENERATED LOCALLY AS DATA IS XMITED
0051   A 0000 000000D1      CRCLOC   EQU     $D1
0052                        *
0053                        *CHMAX IS THE HIGHEST NUMBERED ANALOG CHANNEL
0054   A 0000 0000001F      CHMAX    EQU     $1F
0055                        *
0056                        ********************START OF PROGRAM*********************
0057                        *
0058                        *
0059   A 0000 4F            ANACTR   CLRA
0060   A 0001 B74800                 STA     MUXLT    0 TO MUX LATCH
0061   A 0004 9715                   STA     MUXCT    INIT MUXCT
0062   A 0006 9617                   LDA     STATUS   XMIT STATUS
0063   A 0008 BDFB00                 JSR     LXMIT
0064   A 000B 97D1                   STA     CRCLOC
0065   A 000D 863B                   LDA     #NUMBYT  XMIT NUMBER OF BYTES
0066   A 000F BDFB00                 JSR     LXMIT
0067   A 0012 9BD1                   ADDA    CRCLOC
0068   A 0014 97D1                   STA     CRCLOC
0069   A 0016 8E0007                 LDX     #0007    WAIT FOR ANALOG SETTLING
0070   A 0019 301F          WAIT0    LEAX    -1,X
0071   A 001B 26FC                   BNE     WAIT0
0072   A 001D BDF8E0        CONV0    JSR     CONVRT   CONVERT 0 CHANNEL
0073   A 0020 2D0C                   BLT     TOUT0    TIMED OUT?
0074   A 0022 4F                     CLRA    NO:
0075   A 0023 4C                     INCA
0076   A 0024 B74800                 STA     MUXLT    INCREMENT MUX LATCH
0077   A 0027 BDF890                 JSR     LIMCK    CHECK LIMITS ON PREVIOUS CONV.
0078   A 002A 0C15                   INC     MUXCT    MUXCT+1 TO MUXCT
0079   A 002C 2011                   BRA     ACON0    SKIP OVER YES
0080   A 002E 4F            TOUT0    CLRA    YES:
0081   A 002F 4C                     INCA    INCREMENTMUX LATCH
0082   A 0030 B74800                 STA     MUXLT
0083   A 0033 BDFAE0                 JSR     LIMSET   SET LIMIT BIT
```

```
AMERICAN AUTOMATION ASSEMBLER          ANACTR  20  Dec  83          PAGE 003

0084   A 0036 0C15                     INC     MUXCT
0085   A 0038 8E0002                   LDX     #0002    WAIT FOR ANALOG SETTLING
0086   A 003B 301F          WAIT1      LEAX    -1,X
0087   A 003D 26FC                     BNE     WAIT1
0088   A 003F 1F30          ACON0      TFR     U,D      SHIFT CONVERTED WORD
0089   A 0041 58                       ASLB
0090   A 0042 49                       ROLA
0091   A 0043 58                       ASLB
0092   A 0044 49                       ROLA
0093   A 0045 58                       ASLB
0094   A 0046 49                       ROLA
0095   A 0047 58                       ASLB
0096   A 0048 49                       ROLA
0097   A 0049 D7CF                     STB     TEMP1    KEEP CONVERTED LSB'S
0098   A 004B BDFB00                   JSR     LXMIT    TRANSMIT A DATA BYTE
0099   A 004E 9BD1                     ADDA    CRCLOC
0100   A 0050 97D1                     STA     CRCLOC
0101   A 0052 BDF8E0          CONV1    JSR     CONVRT   CONVERT NEXT CHANNEL
0102   A 0055 2D0D                     BLT     TOUT1    TIMED OUT?
0103   A 0057 9615                     LDA     MUXCT    NO:
0104   A 0059 4C                       INCA    UPDATE   MUXLT
0105   A 005A B74800                   STA     MUXLT
0106   A 005D BDF890                   JSR     LIMCK    CHECK LIMITS
0107   A 0060 0C15                     INC     MUXCT    UPDATE MUXCT
0108   A 0062 200B                     BRA     ACON1    SKIP 'YES'
0109   A 0064 9615          TOUT1      LDA     MUXCT    YES:
0110   A 0066 4C                       INCA    UPDATE   MUXLT
0111   A 0067 B74800                   STA     MUXLT
0112   A 006A BDFAE0                   JSR     LIMSET   SET LIMIT BIT
0113   A 006D 0C15                     INC     MUXCT    UPDATE MUXCT
0114   A 006F 1F30          ACON1      TFR     U,D
0115   A 0071 9ACF                     ORA     TEMP1
0116   A 0073 BDFB00                   JSR     LXMIT    TRANSMIT A DATA BYTE
0117   A 0076 9BD1                     ADDA    CRCLOC
0118   A 0078 97D1                     STA     CRCLOC
0119   A 007A 1F30                     TFR     U,D
0120   A 007C 1F98                     TFR     B,A
0121   A 007E BDFB00                   JSR     LXMIT    TRANSMIT ANOTHER DATA BYTE
0122   A 0081 9BD1                     ADDA    CRCLOC
0123   A 0083 97D1                     STA     CRCLOC
0124   A 0085 BDF8E0                   JSR     CONVRT   CONVERT NEXT CHANNEL
0125   A 0088 2D13                     BLT     TOUT2    TIMED OUT?
```

```
AMERICAN AUTOMATION ASSEMBLER          ANACTR  20  Dec  83          PAGE 004

0126   A 008A 9615                      LDA     MUXCT    NO:
0127   A 008C 4C                        INCA
0128   A 008D B748@@                    STA     MUXLT    UPDATE MUX LATCH
0129   A 0090 811F                      CMPA    #CHMAX   SEE IF DONE
0130   A 0092 2F01                      BLE     ACON2    NO: SKIP
0131   A 0094 39                        RTS     YES:     RETURN
0132   A 0095 BDF89@            ACON2   JSR     LIMCK    CHECK LIMIT
0133   A 0098 0C15                      INC     MUXCT    MUXCT+1 TO MUXCT
0134   A 009A 7E003F      A             JMP     ACON@    LOOP UNTIL DONE
0135   A 009D 9615              TOUT2   LDA     MUXCT    YES:
0136   A 009F 4C                        INCA     INCREMENTMUX LATCH
0137   A 00A0 B748@@                    STA     MUXLT
0138   A 00A3 811F                      CMPA    #CHMAX   SEE IF DONE
0139   A 00A5 2F01                      BLE     ACON3    NO:SKIP
0140   A 00A7 39                        RTS     YES:     RETURN
0141   A 00A8 BDFAE@            ACON3   JSR     LIMSET   SET LIMIT BIT
0142   A 00AB 0C15                      INC     MUXCT    INCREMENT MUXCT
0143   A 00AD 7E003B      A             JMP     WAIT1    LOOP UNTIL DONE
0144                                    END
```

```
AMERICAN AUTOMATION ASSEMBLER          ADUT    20  Dec  83         PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * ADUT SUBROUTINE                                          *
0004                        *                                                          *
0005                        *   FUNCTION: LOADS D/A LATCHES                            *
0006                        *    INPUT:   START ADDR. OF LATCH IN Y                    *
0007                        *             START ADDR. OF DATA IN X                     *
0008                        *             NUMBER OF WORDS IN A                         *
0009                        *   OUTPUT:   DATA TO LATCHES                              *
0010                        *   USES:     A,Y,U,X                                      *
0011                        *                                                          *
0012                        ************************************************************
0013                        *
0014                        *
0015                        ****************START OF PROGRAM**************************
0016                        *
0017  A 0000 EE81           ADUT    LDU     ,X++    GET DATA FROM TABLE
0018  A 0002 EFA1                   STU     ,Y++    STORE TO LATCH
0019  A 0004 4A                     DECA
0020  A 0005 26F9                   BNE     ADJT    LOOP UNTIL DONE
0021  A 0007 39                     RTS
0022                          END
```

```
AMERICAN AUTOMATION ASSEMBLER          CONVRT  20  Dec  83          PAGE 001

0001                        ********************************************************
0002                        *                                                      *
0003                        * CONVERT SUBROUTINE                                   *
0004                        *                                                      *
0005                        *   FUNCTION: GETS THE ADC TO PERFORM A CONVERSION.    *
0006                        *             TIMES OUT IN ABOUT 90 MICROSEC. IF ADC   *
0007                        *             IS NOT READY. UPDATES THE ADC STATUS     *
0008                        *             BIT IN THE STATUS WORD.                  *
0009                        *    INPUT:NONE                                        *
0010                        *    OUTPUT:ADC OUTPUT,MASKED TO 12 BITS IN D,U:UPDATED *
0011                        *           STATUS WORD.                               *
0012                        *   USES:  D,U                                         *
0013                        *                                                      *
0014                        ********************************************************
0015                        *
0016                        *
0017  A 0000 00004801       ADCST   EQU     $4801
0018  A 0000 00004802       ADCOUT  EQU     $4802
0019  A 0000 00000017       STATUS  EQU     $17
0020                        *
0021                        *
0022  A 0000 B74801         CONVRT  STA     ADCST    START CONVERT
0023  A 0003 86FB                   LDA     #$FB     ZERO STATUS BIT
0024  A 0005 9417                   ANDA    STATUS
0025  A 0007 9717                   STA     STATUS
0026  A 0009 CE0003                 LDU     #03      SET UP WAIT LOOP
0027  A 000C 335F           WAITC   LEAU    -1,U     WAIT
0028  A 000E 11830000               CMPU    #0000
0029  A 0012 2F0A                   BLE     TOUTC    BRANCH IF TIMED OUT
0030  A 0014 FC4802                 LDD     ADCOUT   GET ADC WORD
0031  A 0017 2CF3                   BGE     WAITC    0 MSBIT MEANS NOT READY
0032  A 0019 840F                   ANDA    #$0F     MASK OUT TO 12 BITS
0033  A 001B 1F03                   TFR     D,U      PUT RESULT IN U
0034  A 001D 39                     RTS
0035  A 001E 8604           TOUTC   LDA     #04      PUT 1 IN STATUS BIT
0036  A 0020 9A17                   ORA     STATUS
0037  A 0022 9717                   STA     STATUS
0038  A 0024 8A80                   ORA     #$80     FLAG TIME-OUT TO CALLER
0039  A 0026 39                     RTS
0040                                END
```

```
AMERICAN AUTOMATION ASSEMBLER          DIGINT  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * DIGINT SUBROUTINE                                        *
0004                        *                                                          *
0005                        *   FUNCTION: ACQUIRES AND TRANSMITS DIGITAL INPUTS        *
0006                        *   INPUTS:   32 BITS FROM USER                            *
0007                        *   OUTPUTS:  BITS TO LINK UART                            *
0008                        *   USES:     A,Y,LXMIT                                    *
0009                        *                                                          *
0010                        ************************************************************
0011                        *
0012                        *
0013                        ******************SYMBOL DEFINITION*************************
0014                        *
0015                        *LXMIT, A SUBROUTINE, INTERFACES WITH LINK UART
0016  A 0000 0000FB00       LXMIT    EQU     $FB00
0017                        *
0018                        *TSW0 IS THE ADDRESS OF USER BITS 0 - 7
0019  A 0000 0000B000       TSW0     EQU     $B000
0020                        *
0021                        *TSW3 IS THE ADDRESS OF USER BITS 24 - 31
0022  A 0000 0000B003       TSW3     EQU     $B003
0023                        *
0024                        *CRCLOC IS THE CRC TO BE TRANSMITTED
0025  A 0000 000000D1       CRCLOC   EQU     $D1
0026                        *
0027                        *
0028                        ******************START OF PROGRAM**************************
0029                        *
0030  A 0000 10BE2000       DIGINT   LDY     #TSW0    INIT Y REG
0031  A 0004 A6A0           LOOP1    LDA     ,Y+      GET DIGITAL INPUTS
0032  A 0006 BDFB00                  JSR     LXMIT    TRANSMIT DIGITAL INPUTS
0033  A 0009 9BD1                    ADDA    CRCLOC   UPDATE CRC
0034  A 000B 97D1                    STA     CRCLOC
0035  A 000D 10AC3003                CMPY    #TSW3    DONE ?
0036  A 0011 24F1                    BLE     LOOP1    NO,LOOP
0037  A 0013 39                      RTS
0038                           END
```

```
AMERICAN AUTOMATION ASSEMBLER          DISPLY  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * DISPLY SUBROUTINE                                        *
0004                        *   FUNCTION: HANDLES FRONT PANEL AND USER ANALOG          *
0005                        *             MONITOR DISPLAY.                             *
0006                        *   INPUT:    FRONT PANEL ROTARY SWITCH,                   *
0007                        *             FRONT PANEL AND USER DIGISWITCHES            *
0008                        *   OUTPUT:   FRONT PANEL AND USER DISPLAYS                *
0009                        *   USES:     D,X,Y,GET2,DSWRD                             *
0010                        *                                                          *
0011                        ************************************************************
0012                        *
0013                        *
0014                        ******************SYMBOL DEFINITION**************************
0015                        *
0016                        *SELSW IS THE ADDR. OF THE FRONT PANEL ROTARY SEL. SW.
0017  A 0000 0000B000       SELSW    EQU     $B000
0018                        *
0019                        *LRSTER IS THE LOCAL RESET ERROR COUNTER
0020  A 0000 0000C000       LRSTER   EQU     $C000
0021                        *
0022                        *LLATCH IS THE LOCAL LED DISPLAY LATCH
0023  A 0000 00006800       LLATCH   EQU     $6800
0024                        *
0025                        *RLATCH IS THE REMOTE LED DISPLAY LATCH
0026  A 0000 00007000       RLATCH   EQU     $7000
0027                        *
0028                        *RSTER0 IS THE ADDRESS OF THE RESET ERROR IN RBUF0
0029  A 0000 00000059       RSTER0   EQU     $59
0030                        *
0031                        *RSTER1 IS THE ADDRESS OF THE RESET ERROR IN RBUF1
0032  A 0000 000000A9       RSTER1   EQU     $A9
0033                        *
0034                        *RBUFSE IS THE DATA BUF POINTER
0035                        * IF RBUFSE=0, LINT SUBROUTINE IS USING RBUF0
0036                        * IF RBUFSE=1, LINT SUBROUTINE IS USING RBUS1
0037                        * THIS SUBROUTINE USES THE INACTIVE BUFFER
0038  A 0000 000000C1       RBUFSE   EQU     $C1
0039                        *
0040                        *LCOMER IS THE LOCALLY DETECTED COMMUNICATION ERROR COUNT
0041  A 0000 000000D3       LCOMER   EQU     $D3
```

```
AMERICAN AUTOMATION ASSEMBLER          DISPLY  20  Dec  83          PAGE 002

0042                                   *
0043                                   *COMER0 IS THE RBUF0 ADDR. OF COMM. ERROR
0044   A 0000 00000058                 COMER0   EQU     $58
0045                                   *
0046                                   *COMER1 IS THE RBUF1 ADDR. OF COMM. ERROR
0047   A 0000 000000A8                 COMER1   EQU     $A8
0048                                   *
0049                                   *STATUS IS THE LOCAL STATUS WORD
0050   A 0000 00000017                 STATUS   EQU     $17
0051                                   *
0052                                   *REMSTA IS THE REMOTE STATUS WORD
0053   A 0000 000000C9                 REMSTA   EQU     $C9
0054                                   *
0055                                   *BLKDIS IS USED TO KEEP UP WITH DISPLAY BLANKING
0056                                   * BIT 0 IS USED TO BLANK THE FRONT PAN. DVM
0057                                   * BIT 1 IS USED TO BLANK THE REMOTE LED STATUS DISPLAY
0058                                   * THE DVM DISPLAY IS BLINKED WHEN BIT=0
0059                                   * THE LED DISPLAY IS BLANKED WHEN BIT=1
0060   A 0000 000000D5                 BLKDIS   EQU     $D5
0061                                   *
0062                                   *DOUT4 IS THE FOURTH DIGITAL OUTPUT BYTE
0063   A 0000 00006012                 DOUT4    EQU     $6012
0064                                   *
0065                                   *RBUF0 IS THE FIRST ADDRESS IN A RECEIVED REMOTE
0066                                   *  DATA BUFFER
0067   A 0000 00000020                 RBUF0    EQU     $20
0068                                   *
0069                                   *RBUF1 IS THE FIRST ADDRESS IN A RECEIVED REMOTE
0070                                   *  DATA BUFFER
0071   A 0000 00000070                 RBUF1    EQU     $70
0072                                   *
0073                                   *DSW0 IS THE ADDRESS OF THE FRONT PANEL DIGISWITCH
0074                                   *  ASSOCIATED WITH THE FRONT PAN. DVM
0075   A 0000 00008000                 DSW0     EQU     $8000
0076                                   *
0077                                   *DSW2 IS THE ADDRESS OF THE USER DIGISWITCH ASSOCIATED
0078                                   *  WITH DSW2 ENABLE
0079   A 0000 00009000                 DSW2     EQU     $9000
0080                                   *
0081                                   *AO29 IS THE LATCH CONTAINING THE LSB OF ANOUT 2-8 AND
0082                                   *  THE MSB OF ANOUT 2-9
0083   A 0000 0000600D                 AO29     EQU     $600D
```

```
AMERICAN AUTOMATION ASSEMBLER          DISPLY  20  Dec  83          PAGE 003

0084                                   *
0085                                   *AO2B IS THE LATCH CONTAINING THE MSB OF ANOUT 2-8
0086   A 0000 0000600C                 AO2B    EQU     $600C
0087                                   *
0088                                   *AO2B IS THE LATCH CONTAINING THE LSB OF ANOUT 2-10 AND
0089                                   *  THE MSB OF ANOUT 2-11
0090   A 0000 00006010                 AO2B    EQU     $6010
0091                                   *
0092                                   *AO2A IS THE LATCH CONTAINING THE MSB OF ANOUT 2-10
0093   A 0000 0000600F                 AO2A    EQU     $600F
0094                                   *
0095                                   *RCVDGD FLAGS THE FIRST GOOD DATA FRAME
0096   A 0000 000000D7                 RCVDGD  EQU     $D7
0097                                   *
0098                                   *GET2 IS A SUBROUTINE WHICH GETS TWO ANALOG VOLTAGES
0099   A 0000 0000FB88                 GET2    EQU     $FB88
0100                                   *
0101                                   *LIMOFF IS THE OFFSET OF THE LIMIT WORD IN THE
0102                                   * DATA TABLES.
0103   A 0000 00000024                 LIMOFF  EQU     $24
0104                                   *
0105                                   *TEMP1 IS A TEMPORARY STORAGE LOCATION
0106   A 0000 000000CF                 TEMP1   EQU     $CF
0107                                   *
0108                                   *DSWRD IS A SUBROUTINE WHICH RETURNS THE BINARY
0109                                   *VALUE OF THE DIGISWITCH REQUESTED BY THE X REGISTER
0110   A 0000 0000F170                 DSWRD   EQU     $F170
0111                                   *
0112                                   *
0113                                   ********************START OF PROGRAM**********************
0114                                   *
0115   A 0000 8603             DISPLY  LDA     #03     READ ROTARY SEL. SW.
0116   A 0002 B4B000                   ANDA    SELSW
0117   A 0005 2738                     BEQ     DISSTA  DISPLAY STATUS IF = 0
0118   A 0007 44                       LSRA
0119   A 0008 271B                     BEQ     DISCOM  DISPLAY COMERR IF =1
0120   A 000A B63000           DISRST  LDA     LRSTER  DISPLAY RESET ERR. IF = 2
0121   A 000D B76800                   STA     LLATCH
0122   A 0010 0DD7                     TST     RCVDGD  ANY REMOTE DATA RECEIVED?
0123   A 0012 273B                     BEQ     BLANKR  IF NOT GO BLANK REMOTE DISP.
0124   A 0014 8E0059                   LDX     #RSTER0 IS SO FIND AND DISPLAY RSTERR
0125   A 0017 0DC1                     TST     RBUFSE
```

```
AMERICAN AUTOMATION ASSEMBLER          DISPLY  20  Dec  83          PAGE 004

0126   A 0019 2603                     BNE     LOAD1
0127   A 001B 8E00A9                   LDX     #RSTER1
0128   A 001E A684          LOAD1      LDA     0,X
0129   A 0020 B77000                   STA     RLATCH
0130   A 0023 2004                     BRA     CONT1
0131   A 0025 96D3          DISCOM     LDA     LCOMER   DISPLAY LOCAL COMM. ERR.
0132   A 0027 B76800                   STA     LLATCH
0133   A 002A 0DD7                     TST     RCVDGD   ANY REMOTE DATA RECEIVED?
0134   A 002C 2721                     BEQ     BLANKR   IF NOT GO BLANK REMOTE DISPLAY
0135   A 002E 8E0058                   LDX     #COMER0
0136   A 0031 0DC1                     TST     RBUFSE   IF SO, FIND AND DISPLAY REMOTE COM. ERR.
0137   A 0033 2603                     BNE     LOAD2
0138   A 0035 8E00A8                   LDX     #COMER1
0139   A 0038 A684          LOAD2      LDA     0,X
0140   A 003A B77000                   STA     RLATCH
0141   A 003D 201A                     BRA     CONT1
0142   A 003F 9617          DISSTA     LDA     STATUS   DISPLAY STATUS
0143   A 0041 B76800                   STA     LLATCH
0144   A 0044 0DD7                     TST     RCVDGD
0145   A 0046 2707                     BEQ     BLANKR   SEE IF ANY REMOTE DATA RECEIVED
0146   A 0048 96C9                     LDA     REMSTA   IF SO, DISPLAY
0147   A 004A B77000                   STA     RLATCH
0148   A 004D 200A                     BRA     CONT1
0149                        * IF NOT, BLANK REMOTE DISPLY AND UNBLINK DVM
0150   A 004F 8603          BLANKR     LDA     #03
0151   A 0051 9AD5                     ORA     BLKDIS
0152   A 0053 97D5                     STA     BLKDIS
0153   A 0055 B76012                   STA     DOUT4
0154   A 0058 39                       RTS
0155   A 0059 86FD          CONT1      LDA     #$FD     UNBLANK DISPLAY
0156   A 005B 94D5                     ANDA    BLKDIS
0157   A 005D 97D5                     STA     BLKDIS
0158   A 005F B76012                   STA     DOUT4
0159   A 0062 108E00E0      CONT3      LDY     #RBUF0   TABLE ADDR. TO Y
0160   A 0066 0DC1                     TST     RBUFSE
0161   A 0068 2604                     BNE     CONT4
0162   A 006A 108E0070                 LDY     #RBUF1
0163   A 006E 109FCF        CONT4      STY     TEMP1    SAVE RBUF VAL. FOR LATER
0164   A 0071 8E8000                   LDX     #DSW0    DSW ADDR. TO X
0165   A 0074 BDF170                   JSR     DSWRD    GET SW VALUE IN BINARY
0166   A 0077 1F89          CONT6      TFR     A,B      FIND LIMIT BIT
0167   A 0079 44                       LSRA
```

```
AMEFICAN AUTOMATION ASSEMBLER          DISPLY  20  Dec  83          PAGE 005

0168   A 007A 44                       LSRA
0169   A 007B 44                       LSRA
0170   A 007C 8B34                     ADDA    #LIMOFF
0171   A 007E 31A6                     LEAY    A,Y       ADDR. OF LIM WORD TO Y
0172   A 0080 C407                     ANDB    #07       FIND THE BIT WITHIN THE WORD
0173   A 0082 5C                       INCB
0174   A 0083 4F                       CLRA
0175   A 0084 1A01                     ORCC    01
0176   A 0086 46            LOOP1      RORA
0177   A 0087 5A                       DECB
0178   A 0088 26FC                     BNE     LOOP1     LOOP TO POSITION BIT
0179   A 008A A5A4                     BITA    0,Y
0180   A 008C 2706                     BEQ     OK1       BRANCH IF WITHIN LIMITS
0181   A 008E 86FE          TOOBIG     LDA     #$FE
0182   A 0090 94D5                     ANDA    BLKDIS    OUT OF RANGE, BLINK DVM
0183   A 0092 2004                     BRA     CONT7
0184   A 0094 8601          OK1        LDA     #$01      IN RANGE, DON'T BLINK DVM
0185   A 0096 9AD5                     ORA     BLKDIS
0186   A 0098 97D5          CONT7      STA     BLKDIS
0187   A 009A B7E012                   STA     DOUT4
0188   A 009D 10AECF                   LDY     TEMP1     RETRIEVE RBUF VALUE
0189   A 00A0 BDFB88                   JSR     GET2
0190   A 00A3 FDE00D                   STD     AO29      OUTPUT BOTH FRONT PAN. VOLTAGES
0191   A 00A6 96CF                     LDA     TEMP1
0192   A 00A8 B7E00C                   STA     AO28
0193   A 00AB 8E9000                   LDX     #DSW2     USER DSW ADDR. TO X
0194   A 00AE BDFB88                   JSR     GET2      GET REQUESTED VOLTAGES
0195   A 00B1 FDE010                   STD     AO2B      OUTPUT USER MONITOR VOLTAGES
0196   A 00B4 96CF                     LDA     TEMP1
0197   A 00B6 B7E00F                   STA     AO2A
0198   A 00B9 39                       RTS
0199                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER          DSWRD   20  Dec  83          PAGE 001

0001                        ********************************************************
0002                        *                                                      *
0003                        * DSWRD SUBROUTINE                                     *
0004                        *                                                      *
0005                        *   FUNCTION: GETS INPUT FROM DIGISWITCH, CONVERTS IT  *
0006                        *             TO BINARY, FORCES IT TO $1F IF NOT IN THE*
0007                        *             RANGE 0 TO 1F                            *
0008                        *   INPUT:    DIGISWITCH ADDR, IN X                    *
0009                        *             DATA FROM SPECIFIED DIGISWITCH           *
0010                        *   OUTPUT:   DESIRED SWITCH READING IN A REG.         *
0011                        *   USES :    A,B,X                                    *
0012                        *   CALLS:    NONE                                     *
0013                        *   CALLED BY GET2,DISPLY                              *
0014                        *                                                      *
0015                        ********************************************************
0016                        *
0017                        *
0018                        ****************SYMBOL DEFINITION***********************
0019                        *
0020                        *TEN IS THE DECIMAL NUMBER TEN
0021  A 0000 0000000A       TEN     EQU     10
0022                        *
0023                        *MAXBCD IS THE HIGHEST BCD SWITCH VAL. PERMITTED
0024  A 0000 00000031       MAXBCD  EQU     $31
0025                        *
0026                        *MAXBIN IS THE HIGHEST BINARY VALUE PERMITTED
0027  A 0000 0000001F       MAXBIN  EQU     $1F
0028                        *
0029  A 0000 00000000   N           ENT     DSWRD
0030                        *
0031                        *
0032                        ****************START OF PROGRAM************************
0033                        *
0034  A 0000 A684           DSWRD   LDA     0,X     INPUT DIGISWITCH
0035  A 0002 43                     COMA    UNCOMPLEMENT SWITCH
0036  A 0003 8131                   CMPA    #MAXBCD
0037  A 0005 2211                   BHI     OUT     RETURN MAX VAL. IF OUT OF RANGE
0038  A 0007 1F89                   TFR     A,B     BCD TO BINARY CONVERT
0039  A 0009 840F                   ANDA    #$0F    BY SUCCESSIVE ADDITION
0040  A 000B 54                     LSRB    GET     MSD IN B
0041  A 000C 54                     LSRB
0042  A 000D 54                     LSRB
0043  A 000E 54                     LSRB
0044  A 000F 2601                   BNE     LOOP1
0045  A 0011 39                     RTS
0046  A 0012 8B0A           LOOP1   ADDA    #TEN    DO SUCCESSIVE ADDS
0047  A 0014 5A                     DECB
0048  A 0015 26FB                   BNE     LOOP1
0049  A 0017 39                     RTS
0050  A 0018 861F           OUT     LDA     #MAXBIN NO: SET A= MAX
0051  A 001A 39                     RTS
0052                                END
```

```
AMERICAN AUTOMATION ASSEMBLER          ENDTR   20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * ENDTR SUBROUTINE                                         *
0004                        *                                                          *
0005                        *   FUNCTION: HANDLES THE TRANSMISSION OF THE MIS-         *
0006                        *             CELLANEOUS INFORMATION AT THE END OF THE     *
0007                        *             DATA FRAME. THIS INCLUDES THE ANALOG         *
0008                        *             LIMIT WORDS, THE COMMUNICATION ERROR         *
0009                        *             COUNT, THE RESTART ERROR COUNT AND THE       *
0010                        *             CRC WORD.                                    *
0011                        *   INPUT:    NIRTBL,COMERR,RSTERR,CRCLOC                  *
0012                        *   OUTPUT:   ABOVE DATA TO LINK. LINK TIMEOUT CAN         *
0013                        *             CHANGE STATUS WORD.                          *
0014                        *   USES:     A,Y,LXMIT                                    *
0015                        *                                                          *
0016                        ************************************************************
0017                        *
0018                        *
0019                        ******************SYMBOL DEIFINTION*************************
0020                        *
0021                        *NIRTBL IS THE ADDRESS OF THE FIRST ENTRY IN A 4 BYTE
0022                        *   TABLE CONTAINING ANALOG LIMIT BITS.
0023  A 0000 00000010       NIRTBL   EQU     $10
0024                        *
0025                        *NIREND IS THE ADDRESS IF THE LAST ENTRY IN THIS TABLE
0026  A 0000 00000013       NIREND   EQU     $13
0027                        *
0028                        *LCOMER IS USED TO COUNT COMMUNICATION ERRORS
0029  A 0000 000000D3       LCOMER   EQU     $D3
0030                        *
0031                        *RSTER IS A HARDWARE COUNTER TO COUNT RESTART ERRORS
0032  A 0000 00003000       RSTER    EQU     $3000
0033                        *
0034                        *CRCLOC IS THE LOCALLY GENERATED TRANSMIT CRC
0035  A 0000 000000D1       CRCLOC   EQU     $D1
0036                        *
0037                        *LXMIT IS THE LINK TRANSMIT SUBROUTINE
0038  A 0000 0000FB00       LXMIT    EQU     $FB00
0039                        *
0040                        *
0041                        ******************START OF PROGRAM**************************
```

```
AMERICAN AUTOMATION ASSEMBLER          ENDTR  20 Dec 83        PAGE 002

0042                                *
0043   A 0000 108E0010      ENDTR  LDY    #NIRTBL  INIT Y
0044   A 0004 A6A0          LOOP1  LDA    ,Y+      FETCH LIMIT WORDS
0045   A 0006 BDFB00               JSR    LXMIT    XMIT LIMIT WORDS
0046   A 0009 9BD1                 ADDA   CRCLOC   UPDATE CRC
0047   A 000B 97D1                 STA    CRCLOC
0048   A 000D 108C0013             CMPY   #NIREND  DONE?
0049   A 0011 2FF1                 BLE    LOOP1    NO: LOOP
0050   A 0013 96D3                 LDA    LCOMER   YES: CONTINUE
0051   A 0015 BDFB00               JSR    LXMIT    XMIT LCOMER
0052   A 0018 9BD1                 ADDA   CRCLOC
0053   A 001A 97D1                 STA    CRCLOC   UPDATE CRC
0054   A 001C B6C000               LDA    RSTER
0055   A 001F BDFB00               JSR    LXMIT    XMIT RSTER
0056   A 0022 9BD1                 ADDA   CRCLOC   UPDATE CRC
0057   A 0024 BDFB00               JSR    LXMIT    XMIT CRC
0058   A 0027 39                   RTS
0059                               END
```

```
AMERICAN AUTOMATION ASSEMBLER          GET2    20  Dec  83          PAGE 001

0001                    ************************************************************
0002                    *                                                          *
0003                    *   GET2 SUBROUTINE                                        *
0004                    *                                                          *
0005                    *   FUNCTION: READS TWO CONSECUTIVE DIGISWITCHES AND       *
0006                    *             FETCHES THE REQUESTED ANALOG VOLTAGES.       *
0007                    *   INPUT:    RECEIVED DATA TABLE ADDRESS, IN Y:           *
0008                    *             DIGISWITCH ADDRESS IN X                      *
0009                    *   OUTPUT:   FIRST BYTE IN TEMP1, SECOND AND THIRD        *
0010                    *             BYTES IN D REG.                              *
0011                    *   USES:     D,X,Y                                        *
0012                    *                                                          *
0013                    ************************************************************
0014                    *
0015                    *
0016                    ****************SYMBOL DEFINITION*************************
0017                    *
0018                    *TEMP1 IS A TEMPORARY STORAGE LOCATION
0019  A 0000 000000CF   TEMP1   EQU    $CF
0020                    *
0021                    *DSWINC IS THE INCREMENT IN ADDRESS BETWEEN DIGISWITCHES
0022  A 0000 00000600   DSWINC  EQU    $0600
0023                    *
0024                    *T1PLUS IS A TEMPORARY STORAGE LOCATION IMMEDIATELY
0025                    *      FOLLOWING TEMP1.
0026  A 0000 000000D0   T1PLUS  EQU    $D0
0027                    *
0028                    *DSWRD IS A SUBROUTINE WHICH READS A DIGISWITCH AND
0029                    * FORCES IT TO $1F IF IT IS NOT IN THE RANGE 00 TO
0030                    * 31. DECIMAL.
0031  A 0000 0000F170   DSWRD   EQU    $F170
0032                    *
0033                    ****************START OF PROGRAM**************************
0034                    *
0035  A 0000 BDF170     GET2    JSR    DSWRD    READ DIGISW INTO A
0036  A 0003 C603       EVEN1   LDB    #03      UNPACK DATA
0037  A 0005 46                 RORA   SEE      IF EVEN OR ODD
0038  A 0006 840F               ANDA   #$0F
0039                    *ABOVE MASKS IN CASE CARRY WAS SET BEFORE ROTATE
0040                    *AND INSTRUCTION HAS NO EFFECT ON CARRY!
0041  A 0008 240E               BCC    EVEN1    BRANCH ON EVEN
```

```
AMERICAN AUTOMATION ASSEMBLER          GET2    20  Dec  83          PAGE 002

0042   A 000A 3D                       MUL     6A+1     INTO D
0043                          *DON'T LOOK FOR CARRY INTO A SINCE PRODUCT MUST BE <$BB
0044   A 000B 5C                       INCB    DON'T    LOOK FOR CARRY INTO A
0045   A 000C ECAB                     LDD     D,Y      FETCH DATA
0046   A 000E 58                       LSLB    LEFT     JUSTIFY
0047   A 000F 49                       ROLA
0048   A 0010 58                       LSLB
0049   A 0011 49                       ROLA
0050   A 0012 58                       LSLB
0051   A 0013 49                       ROLA
0052   A 0014 58                       LSLB
0053   A 0015 49                       ROLA
0054   A 0016 2005                     BRA     STEMP    NOW GO STORE
0055   A 0018 3D                EVEN1  MUL     6A       INTO D
0056   A 0019 ECAB                     LDD     D,Y      FETCH DATA
0057   A 001B C4F0                     ANDB    #$F0     LEFT JUSTIFY
0058   A 001D DDCF              STEMP  STD     TEMP1    STORE DATA
0059   A 001F 308900800                LEAX    DSWINC,X FETCH NEXT DIGISW
0060   A 0023 BDF170                   JSR     DSWRD    READ NEXT DIGISW.
0061   A 0026 C603              EVENT2 LDB     #03      SEE IF EVEN
0062   A 0028 46                       RORA
0063   A 0029 840F                     ANDA    #$0F     MASK HI NIBBLE
0064   A 002B 2408                     BCC     EVEN2    BRANCH ON EVEN
0065   A 002D 3D                       MUL     6A+1     TO D
0066   A 002E 5C                       INCB
0067                          *RESULT MUST BE <=$BB, SO NO CARRY TO A
0068   A 002F ECAB                     LDD     D,Y      GET DATA
0069   A 0031 840F                     ANDA    #$0F     RIGHT JUSTIFY
0070   A 0033 200B                     BRA     CONT7
0071   A 0035 3D                EVEN2  MUL     6A       TO D
0072   A 0036 ECAB                     LDD     D,Y      FETCH DATA
0073   A 0038 44                       LSRA    RIGHT    JUSTIFY
0074   A 0039 56                       RORB
0075   A 003A 44                       LSRA
0076   A 003B 56                       RORB
0077   A 003C 44                       LSRA
0078   A 003D 56                       RORB
0079   A 003E 44                       LSRA
0080   A 003F 56                       RORB
0081   A 0040 9AD0              CONT7  ORA     T1PLUS
0082                          *THE ABOVE INSTRUCTION PACKS THE DATA FOR OUTPUT
0083                          *TO THE DAC'S. FIRST BYTE IS LEFT IN TEMP1, AND
0084                          *THE SECOND AND THIRD ARE LEFT IN D
0085   A 0042 39                       RTS
0086                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * INIT ROUTINE                                             *
0004                        *                                                          *
0005                        *   FUNCTION: PERFORM SELF-TESTS,AND INITIALIZES PEDAL *
0006                        *   INPUT:    VARIOUS, FROM UNITS TESTED                 *
0007                        *   OUTPUT:   VARIOUS, TO UNITS TESTD AND INITIALIZED    *
0008                        *   CALLS:    NONE                                       *
0009                        *   CALLED BY RESET INTERRUPT                            *
0010                        *                                                          *
0011                        ************************************************************
0012                        *
0013                        *
0014                        ******************SYMBOL DEFINITION**************************
0015                        *
0016                        *UACON IS USER ACIA CONTROL REG.
0017  A 0000 00002800       UACON    EQU     $2800
0018                        *
0019                        *UACIA IS USER ACIA R/W REG
0020  A 0000 00002801       UACIA    EQU     $2801
0021                        *
0022                        *LACON IS LINK ACIA CONTROL REG.
0023  A 0000 00001800       LACON    EQU     $1800
0024                        *
0025                        *LACIA IS LINK ACIA R/W REG.
0026  A 0000 00001801       LACIA    EQU     $1801
0027                        *
0028                        *RAMST IS THE RAM START ADDRESS
0029  A 0000 00000000       RAMST    EQU     $0000
0030                        *
0031                        *RAMLGT IS THE RAM LENGTH
0032  A 0000 00000400       RAMLGT   EQU     $400
0033                        *
0034                        *RAMSTA IS THE STATUS WORD'S MEMORY STATUS BIT
0035  A 0000 00000020       RAMSTA   EQU     $20
0036                        *
0037                        *STATUS IS THE LOCAL STATUS WORD
0038  A 0000 00000017       STATUS   EQU     $17
0039                        *
0040                        *ROMLGT IS THE ROM LENGTH
0041  A 0000 00001000       ROMLGT   EQU     $1000
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20 Dec 93          PAGE 002

0042                                *
0043                                *ROMST IS THE ROM START ADDR.
0044   A 0000 0000F000              ROMST   EQU     $F000
0045                                *
0046                                *CKSUM IS THE STORED ROM CHECKSUM
0047   A 0000 00000000              CKSUM   EQU     $0000
0048                                *
0049                                *ROMSTA IS THE ROM STATUS BIT
0050   A 0000 00000040              ROMSTA  EQU     $40
0051                                *
0052                                *ADCT1 IS AN ADC TEST WORD
0053   A 0000 00000060              ADCT1   EQU     $60
0054                                *
0055                                *ADCT2 IS AN ADC TEST WORD
0056   A 0000 00000040              ADCT2   EQU     $40
0057                                *
0058                                *ADCT3 IS AN ADC TEST WORD
0059   A 0000 00000020              ADCT3   EQU     $20
0060                                *
0061                                *MUXLT IS THE ADC MUX LATCH
0062   A 0000 00004800              MUXLT   EQU     $4800
0063                                *
0064                                *ADOUT1 IS THE FIRST ADC BYTE
0065   A 0000 00004802              ADOUT1  EQU     $4802
0066                                *
0067                                *ADCSTA IS THE ADC STATUS BIT
0068   A 0000 00000004              ADCSTA  EQU     $04
0069                                *
0070                                *CR IS AN ASCII CARRIAGE RETURN
0071   A 0000 0000000D              CR      EQU     $0D
0072                                *
0073                                *PROMPT IS AN ASCII >
0074   A 0000 0000003E              PROMPT  EQU     $3E
0075                                *
0076                                *WAIT1 IS A WAIT COUNT OF ABOUT 6 MILLISECONDS
0077   A 0000 00000600              WAIT1   EQU     $0600
0078                                *
0079                                *WAIT2 IS A WAIT COUNT OF ABOUT 45 MICROSECONDS
0080   A 0000 00000003              WAIT2   EQU     $03
0081                                *
0082                                *UARST IS THE UART RESET WORD
0083   A 0000 00000017              UARST   EQU     $17
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 003

0084                                   *
0085                                   *ACIAS1 SETS UP THE ACIA FOR:
0086                                   * 8 BIT, NO PARITY, 1 STOP, NO INTERRUPTS,/16
0087   A 0000 00000015                 ACIAS1   EQU     $15
0088                                   *
0089                                   *TBEMPT MASKS THE TRANSMIT BUF EMPTY BIT ON ACIA'S
0090   A 0000 00000002                 TBEMPT   EQU     $02
0091                                   *
0092                                   *UACSTA IS THE USER ACIA STATUS BIT
0093   A 0000 00000080                 UACSTA   EQU     $80
0094                                   *
0095                                   *LACSTA IS THE LINK ACIA STATUS BIT
0096   A 0000 00000008                 LACSTA   EQU     $08
0097                                   *
0098                                   *SYMOCT IS THE SYNC MODE COUNTER. INIT VALUE =2
0099   A 0000 0000001B                 SYMOCT   EQU     $1B
0100                                   *
0101                                   *DA0, DA1, DA2 AREA THE D/A CARD ADDRESSES
0102   A 0000 00005000                 DA0      EQU     $5000
0103   A 0000 00005800                 DA1      EQU     $5800
0104   A 0000 00006000                 DA2      EQU     $6000
0105                                   *
0106                                   *DALPS IS THE NUMBER OF LOOPS REQ'D IN DAC INIT
0107   A 0000 00000006                 DALPS    EQU     $06
0108                                   *
0109                                   *TOSTAK IS THE INITIAL TOP OF STACK
0110   A 0000 000003FF                 TOSTAK   EQU     $03FF
0111                                   *
0112                                   *_INTEN SETS LINK ACIA FOR:
0113                                   * 8 BIT, NO PARITY, 1 STOP,/16
0114   A 0000 00000095                 LINTEN   EQU     $95
0115                                   *
0116                                   *UINTEN SETS USER ACIA FOR :
0117                                   * 8 BIT, NO PARITY, 1 STOP, /16
0118   A 0000 00000095                 UINTEN   EQU     $95
0119                                   *
0120                                   *DOFF IS THE OFFSET OF THE DIGITAL WORD IN THE DAC CARD
0121   A 0000 00000012                 DOFF     EQU     $12
0122                                   *
0123                                   *DWRD IS TEH INITIAL DIGITAL WORD OUTPUT TABLE
0124   A 0000 0000F888                 DWRD     EQU     $F888
0125                                   *
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 004

0126                          *INSEL IS THE LINK INPUT CHAN SEL. FLOP
0127   A 0000 00002000        INSEL    EQU     $2000
0128                          *
0129                          *MAIN IS START ADDR. OF MAIN PROG.
0130   A 0000 0000F140        MAIN     EQU     $F140
0131                          *
0132                          *TEMP IS A TEMPORARY STORAGE LOCATION
0133   A 0000 000000CF        TEMP     EQU     $CF
0134                          *
0135                          *R1CK IS THE LOCATION OF THE F000 ROM CHECKSUM
0136   A 0000 0000F7FE        R1CK     EQU     $F7FE
0137                          *
0138                          *R2CK IS THE LOCATION OF THE F800 ROM CHECKSUM
0139   A 0000 0000F881        R2CK     EQU     $F881
0140                          *
0141                          *FROUT IS USED TO COUNT PROGRAM LOOPS WITHOUT A
0142                          *  GOOD FRAME RECEIVED. INITIALIZED TO 1.
0143   A 0000 000000D9        FROUT    EQU     $D9
0144                          *
0145                          ******************START OF PROGRAM************************
0146                          *
0147   A 0000 8617            INIT     LDA     #UARST   DISABLE INTERRUPTS
0148   A 0002 B72800                   STA     UACON
0149   A 0005 B71800                   STA     LACON
0150   A 0008 108E0022    A DACINI    LDY     #DATAB   INIT DAC'S
0151   A 000C CE7FF7                   LDU     #$7FF7   Y GETS DAC ADDR. TABLE
0152   A 000F C6FF                     LDB     #$FF     U,B GET 0 VOLTS BITS
0153   A 0011 AEA4            LOOPA    LDX     0,Y      ADDR. OF DAC TO Y
0154   A 0013 2715                     BEQ     DIGINI   IF ADDR. = 0, THEN DONE
0155   A 0015 8606                     LDA     #DALPS   SET LOOP COUNTER
0156   A 0017 EF81            LOOPB    STU     ,X++     STORE TO DAC'S
0157   A 0019 E780                     STB     ,X+
0158   A 001B 4A                       DECA    DEC      . LOOP COUNTER
0159   A 001C 26F9                     BNE     LOOPB    LOOP UNTIL = 0
0160   A 001E 3122                     LEAY    2,Y      WHEN 0, POINT TO NEXT DAC CARD
0161   A 0020 20EF                     BRA     LOOPA    AND LOOP
0162   A 0022 5000            DATAB    FDB     DA0      TABLE OF DAC ADDRESSES
0163   A 0024 5800                     FDB     DA1
0164   A 0026 6000                     FDB     DA2
0165   A 0028 0000                     FDB     $0000
0166   A 002A 8EF888          DIGINI   LDX     #DWRD    DIGITAL BITS INIT
0167   A 002D 108E0022    A            LDY     #DATAB   TABLE OF INIT. VALUES TO X
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 005

0166   A 0031 EEA4        LOOPC   LDU     0,Y       DAC CARD ADDR TABLE TO Y
0169   A 0033 2709                BEQ     RAMTST    IF 0 TO U THE END OF DATAB
0170   A 0035 EC81                LDD     ,X++      GO TO NEXT TEST
0171   A 0037 EDC812              STD     [#F,U     LOAD AND STORE INIT VALUES
0172   A 003A 3122                LEAY    2,Y       POINT TO NEXT ENTRY IN DATAB
0173   A 003C 20F3                BRA     LOOPC     AND KEEP LOOPING
0174                      *RAM TEST: STORE AND VERIFY TWO BIT PATTERNS
0175   A 003E CCAAAA      RAMTST  LDD     #$AAAA    BIT PATTERN TO D
0176   A 0041 8E0000      LOOP1   LDX     #RAMST    RAM START ADDR. TO X
0177   A 0044 108E0400            LDY     #RAMLGT   RAM LENGTH TO Y
0178   A 0048 ED84        LOOP2   STD     0,X
0179   A 004A 10A381              CMPD    ,X++      STORE AND VERIFY
0180   A 004D 260E                BNE     RAMERR    FLAG ERROR IF NOT VERIFY
0181   A 004F 313E                LEAY    -2,Y
0182   A 0051 26F5                BNE     LOOP2     LOOP THRU ALL RAM
0183   A 0053 10835555            CMPD    #$5555    DONE TWO PATS. YET?
0184   A 0057 270A                BEQ     RAMOK     YES: FLAG RAM OK
0185   A 0059 44                  LSRA      NO:     DO SECOND PATTERN
0186   A 005A 54                  LSRB
0187   A 005B 20E4                BRA     LOOP1
0188   A 005D 8620        RAMERR  LDA     #RAMSTA   ON ERROR, SET STATUS BIT 5
0189   A 005F 9717                STA     STATUS
0190   A 0061 2003                BRA     ROMTST    GO TO NEXT TEST
0191   A 0063 4F          RAMOK   CLRA      NO      ERROR, CLEAR STATUS
0192   A 0064 9717                STA     STATUS
0193                      *ROM TEST: SEE IF SUM OF ALL LOCATIONS=0
0194   A 0066 FCF7FE      ROMTST  LDD     R1CK      GET F000 ROM CHECKSUM
0195   A 0069 F3F881              ADDD    R2CK      SUM WITH F800 ROM CHECKSUM
0196   A 006C DDCF                STD     TEMP      SAVE RESULT
0197   A 006E 108E1000            LDY     #ROMLGT   ROM LENGTH TO Y
0198   A 0072 8EF000              LDX     #ROMST    ROM START TO X
0199   A 0075 4F                  CLRA      SUM     ALL LOCATIONS
0200   A 0076 5F                  CLRB
0201   A 0077 EBA0        LOOP3   ADDB    ,X+
0202   A 0079 2401                BCC     SKIP1     BRANCH IF NO CARRY
0203   A 007B 4C                  INCA
0204   A 007C 313F        SKIP1   LEAY    -1,Y
0205   A 007E 26F7                BNE     LOOP3     LOOP UNTIL DONE
0206   A 0080 1083CF              CMPD    TEMP      SEE IF SUM CHECKS
0207   A 0083 2706                BEQ     ANTST     IF OK GO TO NEXT TEST
0208   A 0085 8640                LDA     #ROMSTA   IF NOT OK SET ROM STATUS BIT
0209   A 0087 9A17                ORA     STATUS
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 006

0210   A 0089 9717                     STA     STATUS
0211                                *ADC BOARD TEST: SEE THAT 2 CONTROL BITS
0212                                * GET IN AND OUT OK
0213   A 008B 4F                    ANTST   CLRA    STORE   0 IN CONTROL BITS
0214   A 008C B74800                        STA     MUXLT
0215   A 008F 8660                          LDA     #ADCT1  SEE IF THEY COME OUT OK
0216   A 0091 B44802                        ANDA    ADOUT1  MASK BITS WE WANT TO SEE
0217   A 0094 8140                          CMPA    #ADCT2  SEE IF GET CORRECT RESULT
0218   A 0096 260E                          BNE     ADERR   IF NOT, FLAG ERROR
0219   A 0098 4F                            CLRA    NOW     STORE ONES
0220   A 0099 43                            COMA
0221   A 009A B74800                        STA     MUXLT
0222   A 009D 8660                          LDA     #ADCT1  MASK BITS
0223   A 009F B44802                        ANDA    ADOUT1
0224   A 00A2 8120                          CMPA    #ADCT3  SEE IF GET CORRECT RESULT
0225   A 00A4 2706                          BEQ     UARTST  YES: GO TO NEXT TEST
0226   A 00A6 8604                  ADERR   LDA     #ADCSTA NO:-LAG ERROR
0227   A 00A8 9A17                          ORA     STATUS
0228   A 00AA 9717                          STA     STATUS
0229                                *ACIATST: SEE IF TRANSMIT BUF GOES FULL
0230                                * AND THEN EMPTY AFTER LOADING A CHARACTER
0231   A 00AC 8E0600                UARTST  LDX     #WAIT1  LOAD WAIT COUNT TO X
0232   A 00AF 8615                          LDA     #ACIAS1 DISABLE ACIA INTERRUPTS
0233   A 00B1 B72800                        STA     UACON
0234   A 00B4 B71800                        STA     LACON
0235   A 00B7 C602                          LDB     #TBEMPT TBUF EMPTY TEST WORD
0236   A 00B9 863E                          LDA     #PROMPT TEST DATA TO SEND
0237   A 00BB B72801                        STA     UACIA   STORE BYTE
0238   A 00BE F52800                        BITB    UACON   TBUF FULL?
0239   A 00C1 2609                          BNE     UACERR  NO: ERROR
0240   A 00C3 F52800                LOOPU   BITB    UACON   YES: WAIT FOR EMPTY
0241   A 00C6 260A                          BNE     LACTST  ON EMPTY GO TEST LACIA
0242   A 00C8 301F                          LEAX    -1,X    MARK TIME
0243   A 00CA 26F7                          BNE     LOOPU   LOOP IF NO TIME OUT
0244   A 00CC 8680                  UACERR  LDA     #UACSTA ON ERROR, SET STATUS BIT
0245   A 00CE 9A17                          ORA     STATUS
0246   A 00D0 9717                          STA     STATUS
0247                                *LINK ACIA TEST
0248   A 00D2 8E0003                LACTST  LDX     #WAIT2  SET UP WAIT COUNT
0249   A 00D5 863E                          LDA     #PROMPT SEND PROMPT
0250   A 00D7 B71801                        STA     LACIA
0251   A 00DA F51800                        BITB    LACON   TBUF FULL?
```

```
AMERICAN AUTOMATION ASSEMBLER          INIT    20  Dec  83          PAGE 007

0252   A 00DD 2609                     BNE     LACERR   NO: ERROR
0253   A 00DF F51800        LOOPL      BITB    LACON    YES: WAIT FOR EMPTY
0254   A 00E2 260A                     BNE     RAMSET   ON EMPTY GO TEST RAM
0255   A 00E4 301F                     LEAX    -1,X     MARK TIME
0256   A 00E6 26F7                     BNE     LOOPL    LOOP IF NO TIME OUT
0257   A 00E8 8608          LACERR     LDA     #LACSTA  NOT OK: FLAG ERROR
0258   A 00EA 9A17                     ORA     STATUS
0259   A 00EC 9717                     STA     STATUS
0260                        *INITIALIZE ALL MEMORY TO ZERO EXCEPT
0261                        * FROUT WHICH GOES TO ONE, AND
0262                        * SYMOCT WHICH GOES TO TWO AND
0263                        * STATUS WHICH IS PRESERVED AS IS
0264   A 00EE 9617          RAMSET     LDA     STATUS   PRESERVE STATUS
0265   A 00F0 1F03                     TFR     D,U
0266   A 00F2 4F                       CLRA    ZERO     TO D
0267   A 00F3 5F                       CLRB
0268   A 00F4 8E0000                   LDX     #RAMST
0269   A 00F7 108E0400                 LDY     #RAMLGT
0270   A 00FB ED81          LOOP5      STD     ,X++     STORE ZEROES
0271   A 00FD 313E                     LEAY    -2,Y     THROUGHOUT MEMORY
0272   A 00FF 26FA                     BNE     LOOP5
0273   A 0101 8601                     LDA     #01      FROUT TO 1
0274   A 0103 97D9                     STA     FROUT
0275   A 0105 8602                     LDA     #02      SYMOCT TO 2
0276   A 0107 971B                     STA     SYMOCT
0277   A 0109 1F30                     TFR     U,D
0278   A 010B 9717                     STA     STATUS   RESTORE STATUS
0279   A 010D 4F            LCHINI     CLRA    INITIALIZE LINK CHAN.
0280   A 010E B72000                   STA     INSEL
0281   A 0111 8617          INTEN      LDA     #UARST   RESET ACIA'S
0282   A 0113 B71800                   STA     LACON
0283   A 0116 B72800                   STA     UACON
0284   A 0119 8601                     LDA     #$01     SIGNAL LOOKING FOR GOOD FRAME
0285   A 011B 9A17                     ORA     STATUS
0286   A 011D 9717                     STA     STATUS
0287   A 011F 8695                     LDA     #LINTEN  READY LINK ACIA
0288   A 0121 B71800                   STA     LACON
0289   A 0124 8695                     LDA     #UINTEN  READY USER ACIA
0290   A 0126 B72800                   STA     UACON
0291   A 0129 1CAF                     ANDC    $AF      REMOVE INTERRUPT MASKS
0292   A 012B 10CE03FF                 LDS     #TOSTAK  ENABLE NMI BY LDS
0293   A 012F 7EF140                   JMP     MAIN

0294                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER        LIMCK   20  Dec  83          PAGE 001

0001                    ********************************************************
0002                    *                                                      *
0003                    *  LIMCK SUBROUTINE                                    *
0004                    *    CHECKS MIN/MAX LIMITS OF ANALOG VOLTAGES          *
0005                    *    INPUT:                                            *
0006                    *          U-REG FOR ANALOG VOLTAGE                    *
0007                    *          MUXCT (LOC 15) TO COMPUTE BIT TO UPDATE     *
0008                    *          RANTBL,A TABLE, CONTAINING MIN/MAX VALUES   *
0009                    *    OUTPUT:                                           *
0010                    *           NIRTBL, A TABLE, GETS ONE BIT SET OR RESET *
0011                    *    ALSO USES:                                        *
0012                    *              A,B,X,Y                                 *
0013                    *                                                      *
0014                    ********************************************************
0015                    *
0016                    *
0017  A 0000 00000010   NIRTBL   EQU     $0010
0018  A 0000 00000015   MUXCT   EQU     $15
0019  A 0000 0000F800   RANTBL  EQU     $F800
0020                    *
0021                    *
0022  A 0000 108E0010   LIMCK   LDY     #NIRTBL  STATUS TABLE ADDR. TO Y
0023  A 0004 4F                 CLRA    CALCULATEADDR IN RANTBL
0024  A 0005 D615               LDB     MUXCT
0025  A 0007 58                 ASLB
0026  A 0008 58                 ASLB
0027  A 0009 C3F800             ADDD    #RANTBL
0028  A 000C 1F01               TFR     D,X       ADDR. IN RANTBL TO X
0029  A 000E 9615               LDA     MUXCT     FIND POSITION OF BIT TO UPDATE
0030  A 0010 8407               ANDA    #07       KEEP ONLY 3 LSB'S
0031  A 0012 4C                 INCA
0032  A 0013 5F                 CLRB
0033  A 0014 1A01               ORCC    01        SHIFT A BIT FROM C THRU B
0034  A 0016 56         LIMSH   RORB
0035  A 0017 4A                 DECA
0036  A 0018 26FC               BNE     LIMSH
0037  A 001A 43                 COMA    SEE       IF LIMITS ARE PROGRAMMED
0038  A 001B A184               CMPA    0,X
0039  A 001D 270A               BEQ     IR        BRANCH IF NOT
0040  A 001F 11A381             CMPU    ,X++      CHECK LIMITS
0041  A 0022 2E10               BGT     NIR
```

```
AMERICAN AUTOMATION ASSEMBLER        LIMCK  20 Dec 83        PAGE 002

0042   A 0024 11A384                 CMPU    0,X
0043   A 0027 2D0B                   BLT     NIR
0044   A 0029 9615          IR       LDA     MUXCT    WITHIN RANGE OR UNPROGRAMMED
0045   A 002B 44                     LSRA    UPDATE   THE APPROPRIATE BIT
0046   A 002C 44                     LSRA
0047   A 002D 44                     LSRA
0048   A 002E 53                     COMB
0049   A 002F E4A6                   ANDB    A,Y
0050   A 0031 E7A6                   STB     A,Y
0051   A 0033 39                     RTS
0052   A 0034 9615          NIR      LDA     MUXCT    NOT WITHIN RANGE
0053   A 0036 44                     LSRA    UPDATE   APPROPRIATE BIT
0054   A 0037 44                     LSRA
0055   A 0038 44                     LSRA
0056   A 0039 EAA6                   ORB     A,Y
0057   A 003B E7A6                   STB     A,Y
0058   A 003D 39                     RTS
0059                                 END
```

```
AMERICAN AUTOMATION ASSEMBLER          LIMSET  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * LIMSET SUBROUTINE                                        *
0004                        *                                                          *
0005                        *   FUNCTION: SETS LIMIT BIT IN NIRTBL, PER MUXCT          *
0006                        *   INPUT:    MUXCT, SPECIFIES WHICH BIT TO SET            *
0007                        *   OUTPUT:   THE SPECIFIED BIT IS SET IN NIRTBL           *
0008                        *   USES:     A,B,X                                        *
0009                        *                                                          *
0010                        ************************************************************
0011                        *
0012                        *
0013                        *NIRTBL IS THE STARTING ADDRESS OF THE 'NOT IN RANGE'      *
0014                        *       TABLE .                                            *
0015  A 0000 00000010       NIRTBL   EQU     $10
0016                        *
0017                        *MUXCT IS USED TO KEEP TRACK OF THE CHANNEL TO CONVERT.*
0018  A 0000 00000015       MUXCT    EQU     $15
0019                        *
0020                        *
0021                        ****************START OF PROGRAM************************
0022                        *
0023                        *
0024  A 0000 8E0010         LIMSET   LDX     #NIRTBL  STATUS TABLE ADDR. TO X
0025  A 0003 9615                    LDA     MUXCT    POSITION BIT
0026  A 0005 8407                    ANDA    #07
0027  A 0007 4C                      INCA
0028  A 0008 5F                      CLRB
0029  A 0009 1A01                    ORCC    #1
0030  A 000B 56             LSSHFT   RORB
0031  A 000C 4A                      DECA
0032  A 000D 26FC                    BNE     LSSHFT
0033  A 000F 9615                    LDA     MUXCT    STORE BIT
0034  A 0011 44                      LSRA
0035  A 0012 44                      LSRA
0036  A 0013 44                      LSRA
0037  A 0014 EA86                    ORB     A,X
0038  A 0016 E786                    STB     A,X
0039  A 0018 39                      RTS
0040                                 END
```

```
AMERICAN AUTOMATION ASSEMBLER          LINT    20  Dec  83          PAGE 001

0001                    ********************************************************
0002                    *                                                      *
0003                    * LINT SUBROUTINE                                      *
0004                    *                                                      *
0005                    *   FUNCTION: HANDLES RECEIVED DATA INTERRUPTS FROM    *
0006                    *             LINK.                                    *
0007                    *   INPUT:    LINK RECEIVED DATA AND MISC. MEMORY      *
0008                    *             LOCATIONS.                               *
0009                    *   OUTPUT:   LINK DATA IN BUFFERS, REMOTE STATUS, AND *
0010                    *             MISC. MEMORY LOCATIONS.                  *
0011                    *   USES:     A,B,X,S-STACK                            *
0012                    *                                                      *
0013                    ********************************************************
0014                    *
0015                    *
0016                    *****************SYMBOL DEFINITIONS*********************
0017                    *
0018                    *LINTCT IS USED TO TELL MAIN MEMORY AN INTERRUPT OCCURED
0019  A 0000 00000019   LINTCT   EQU     $19
0020                    *
0021                    *SYMOCT IS USED TO DEFINE THE SYNC MODE
0022                    *   =3 MEANS IGNORE NEXT DATA BYTE
0023                    *   =2 MEANS LOOK FOR FIRST SYNC WORD
0024                    *   =1 MEANS LOOK FOR SECOND SYNC WORD
0025                    *   =0 MEANS IN SYNC
0026  A 0000 0000001B   SYMOCT   EQU     $1B
0027                    *
0028                    *SYNTBL IS THE ADDRESS OF A TABLE DEFINED BELOW
0029                    *
0030                    *RCVTBL IS THE ADDRESS OF A JUMP TABLE DEFINED BELOW
0031                    *
0032                    *HDRCT IS USED AS A POINTER IN THE DATA FRAME
0033                    *   =4 MEANS LOOK FOR FIRST SYNC WORD
0034                    *   =3 MEANS LOOK FOR SECOND SYNC WORD
0035                    *   =2 MEANS LOOK FOR STATUS WORD
0036                    *   =1 MEANS LOOK FOR BYTE COUNT
0037                    *   =0 MEANS LOOK FOR DATA OR CRC
0038  A 0000 0000001D   HDRCT    EQU     $1D
0039                    *
0040                    *RCVDAT IS THE ADDRESS OF THE LINK UART DATA BUF
0041  A 0000 00001801   RCVDAT   EQU     $1801
```

```
0042                        *
0043                        *BYTECT IS THE DATA BYTE COUNTER
0044   A 0000 0000001F      BYTECT   EQU     $1F
0045                        *
0046                        *RBUF0 IS THE STARTING ADDRESS OF A DATA BUF.
0047   A 0000 00000020      RBUF0    EQU     $20
0048                        *RBUF1 IS THE STARTING ADDRESS OF A DATA BUF.
0049   A 0000 00000070      RBUF1    EQU     $70
0050                        *
0051                        *RBUFSE IS A POINTER TO THE ACTIVE DATA BUFFER
0052   A 0000 000000C1      RBUFSE   EQU     $C1
0053                        *
0054                        *CRCCOM IS USED TO STORE THE COMPUTED CRC
0055   A 0000 000000C3      CRCCOM   EQU     $C3
0056                        *
0057                        *BCRCCT IS USED TO COUNT THE NUMBER OF BAD CRC'S
0058   A 0000 000000C5      BCRCCT   EQU     $C5
0059                        *
0060                        *REMSTA IS USED TO STORE THE REMOTE STATUS
0061   A 0000 000000C9      REMSTA   EQU     $C9
0062                        *
0063                        *FSYNOK IS USED TO FLAG A GOOD FIRST SYNC
0064   A 0000 000000CD      FSYNOK   EQU     $CD
0065                        *
0066                        *STATUS IS USED TO STORE THE LOCAL STATUS
0067   A 0000 00000017      STATUS   EQU     $17
0068                        *
0069                        *BSYNCT IS USED TO COUNT THE BAD SYNC WORDS
0070   A 0000 000000CD      BSYNCT   EQU     $CD
0071                        *
0072                        *FRRDY IS USED TO TELL THE MAIN PROGRAM THAT A
0073                        *COMPLETE, GOOD FRAME HAS BEEN RECEIVED.
0074   A 0000 000000C7      FRRDY    EQU     $C7
0075                        *
0076                        *LCOMER COUNTS COMMUNICATION ERRORS
0077   A 0000 000000D3      LCOMER   EQU$D3
0078                        *
0079                        *RCVPTR IS AN INDEX USED IN STORING RECEIVED DATA
0080   A 0000 000000DF      RCVPTR   EQU     $DF
0081                        *
0082                        *UPDATE IS A SUBROUTINE THAT UPDATES THE ANALOG AND
0083                        *   DIGITAL OUTPUTS AFTER RECEIPT OF A GOOD FRAME.
```

```
AMERICAN AUTOMATION ASSEMBLER          LINT    20 Dec 83        PAGE 003

0084   A 0000 0000FDF0      UPDATE   EQU     $FDF0
0085                        *
0086                        *
0087                        **********START OF PROGRAM******************************
0088                        *
0089                        *
0090   A 0000 001C        A SYNTBL   FDB     SYNC0
0091   A 0002 0024        A          FDB     SYNC1
0092   A 0004 0039        A          FDB     SYNC2
0093   A 0006 0045        A          FDB     SYNC3
0094   A 0008 004A        A RCVTBL   FDB     HDRCT0
0095   A 000A 00B8        A          FDB     HDRCT1
0096   A 000C 00CA        A          FDB     HDRCT2
0097   A 000E 00DE        A          FDB     HDRCT3
0098   A 0010 0100        A          FDB     HDRCT4
0099   A 0012 1A50          LINT     ORCC    $50       MASK INTERRUPTS
0100   A 0014 961B                   LDA     SYMOCT    JUMP PER SYNC MODE TABLE
0101   A 0016 48                     ASLA
0102   A 0017 8E0000      A          LDX     #SYNTBL
0103   A 001A 6E96                   JMP     [A,X]
0104   A 001C 8E0008      A SYNC0    LDX     #RCVTBL   JUMP PER RCV TABLE
0105   A 001F 961D                   LDA     HDRCT
0106   A 0021 48                     ASLA
0107   A 0022 6E96                   JMP     [A,X]
0108   A 0024 B61901        SYNC1    LDA     RCVDAT    SECOND SYNC?
0109   A 0027 817E                   CMPA    #$7E
0110   A 0029 2609                   BNE     NOSYN2    BRANCH IF NOT
0111   A 002B 0A1B                   DEC     SYMOCT    SYNC ACHIEVED
0112   A 002D 8602                   LDA     #02       UPDATE COUNTERS
0113   A 002F 971D                   STA     HDRCT
0114   A 0031 7E0127      A          JMP     OUT
0115   A 0034 0C1B          NOSYN2   INC     SYMOCT    GO LOOK FOR FIRST SYNC
0116   A 0036 7E0127      A          JMP     OUT
0117   A 0039 B61801        SYNC2    LDA     RCVDAT
0118   A 003C 817E                   CMPA    #$7E
0119   A 003E 2602                   BNE     NOSYN1    NO, KEEP LOOKING
0120   A 0040 0A1B                   DEC     SYMOCT    YES,LOOK FOR SECOND
0121   A 0042 7E0127      A NOSYN1   JMP     OUT
0122   A 0045 0A1B          SYNC3    DEC     SYMOCT    SKIP THIS DATA BYTE
0123   A 0047 7E0127      A          JMP     OUT
0124   A 004A B61801        HDRCT0   LDA     RCVDAT    GET DATA
0125   A 004D 0D1F                   TST     BYTECT    CRC?
```

```
AMERICAN AUTOMATION ASSEMBLER          LINT    20  Dec  83          PAGE 004

0126   A 004F 2719                     BEQ     CRCTST   YES,GO TEST CRC
0127   A 0051 D6DF                     LDB     RCVPTR   RCV'D DATA STORE INDEX TO B
0128   A 0053 8E0020                   LDX     #RBUF0   NO,SELECT BUF FOR DATA STORE
0129   A 0056 0DC1                     TST     RBUFSE
0130   A 0058 2703                     BEQ     STRCV
0131   A 005A 8E0070                   LDX     #RBUF1
0132   A 005D A785           STRCV     STA     B,X      STORE DATA
0133   A 005F 0CDF                     INC     RCVPTR   INC THE INDEX
0134   A 0061 0A1F                     DEC     BYTECT   UPDATE BYTECT
0135   A 0063 9BC3                     ADDA    CRCCOM   UPDATE CRC
0136   A 0065 97C3                     STA     CRCCOM
0137   A 0067 7E0127         A         JMP     OUT
0138   A 006A 91C3           CRCTST    CMPA    CRCCOM   RCV'D CRC=COMPUTED?
0139   A 006C 271C                     BEQ     GCRC     YES BRANCH
0140   A 006E 8602           BCRC      LDA     #02      SEE IF <2 BAD
0141   A 0070 91C5                     CMPA    BCRCCT
0142   A 0072 2E0D                     BGT     LT2BAD   YES BRANCH
0143   A 0074 8603                     LDA     #03      NO, RESYNC
0144   A 0076 971B                     STA     SYMDCT
0145   A 0078 8601                     LDA     #01      SET STATUS BIT
0146   A 007A 9A17                     ORA     STATUS
0147   A 007C 9717                     STA     STATUS
0148   A 007E 7E0127         A         JMP     OUT
0149   A 0081 0CC5           LT2BAD    INC     BCRCCT   UPDATE BAD CRC COUNT
0150   A 0083 8604                     LDA     #04      SET UP TO RECEIVE SYNC
0151   A 0085 971D                     STA     HDRCT    (DON'T FLAG DATA READY)
0152   A 0087 7E0127         A         JMP     OUT
0153   A 008A 8604           GCRC      LDA     #04      GOOD CRC
0154   A 008C 971D                     STA     HDRCT    UPDATE HDRCT
0155   A 008E 0FC5                     CLR     BCRCCT   CLEAR BAD CRC COUNTER
0156   A 0090 9617                     LDA     STATUS
0157   A 0092 84FE                     ANDA    #$FE     CLEAR 'LOOKING FOR SYNC'
0158   A 0094 9717                     STA     STATUS   STATUS BIT
0159   A 0096 0DC1                     TST     RBUFSE   UPDATE BUFFER POINTER
0160   A 0098 270F                     BEQ     SE0
0161   A 009A 0FC1                     CLR     RBUFSE
0162   A 009C 0DCD                     TST     BSYNCT   UPDATE DATA?
0163   A 009E 2606                     BNE     NONEW1   NO: GO JUMP OUT
0164   A 00A0 8E0070                   LDX     #RBUF1   YES: POINT TO DATA
0165   A 00A3 BDFDF0                   JSR     UPDATE   GO UPDATE
0166   A 00A6 7E0127         A NONEW1  JMP     OUT
0167   A 00A9 0CC1           SE0       INC     RBUFSE
```

```
AMERICAN AUTOMATION ASSEMBLER          LINT    20  Dec  83          PAGE 005

0168   A 00AB 0DCD                     TST     BSYNCT   UPDATE DATA?
0169   A 00AD 2606                     BNE     NONEW2
0170   A 00AF 8E0020                   LDX     #RBUF0   YES: POINT TO DATA
0171   A 00B2 BDFDF0                   JSR     UPDATE   UPDATE
0172   A 00B5 7E0127       A NONEW2    JMP     OUT
0173   A 00B8 B61801         HDRCT1    LDA     RCVDAT
0174   A 00BB 4A                       DECA    STORE    BYTE COUNT
0175   A 00BC 971F                     STA     BYTECT
0176   A 00BE 4C                       INCA    UPDATE   CRC
0177   A 00BF 9BC3                     ADDA    CRCCOM
0178   A 00C1 97C3                     STA     CRCCOM
0179   A 00C3 0A1D                     DEC     HDRCT    UPDATE HDRCT
0180   A 00C5 0FDF                     CLR     RCVPTR   INITILIZE RCV BUF POINTER
0181   A 00C7 7E0127       A           JMP     OUT
0182   A 00CA B61801         HDRCT2    LDA     RCVDAT   RECEIVE STATUS
0183   A 00CD 97C9                     STA     REMSTA
0184   A 00CF 0A1D                     DEC     HDRCT    UPDATE HDRCT
0185   A 00D1 97C3                     STA     CRCCOM   INIT CRC COUNT
0186   A 00D3 2052                     BRA     OUT
0187   A 00D5 B61801         HDRCT3    LDA     RCVDAT   SECOND SYNC?
0188   A 00D8 817E                     CMPA    #$7E
0189   A 00DA 2604                     BNE     BSYN3    NO,BRANCH
0190   A 00DC 0A1D                     DEC     HDRCT    YES:
0191   A 00DE 0DCD                     TST     FSYNOK   WAS FIRST SYNC OK?
0192   A 00E0 2602                     BNE     FSYNBD   NO,BRANCH OUT
0193   A 00E2 0FCD                     CLR     BSYNCT   YES CLEAR BAD SYNC COUNTER
0194   A 00E4 2041           FSYNBD    BRA     OUT
0195   A 00E6 0CD8           BSYN3     INC     LCUMER
0196   A 00E8 8602                     LDA     #02      SEE HOW MANY BAD SYNCS
0197   A 00EA 91CD                     CMPA    BSYNCT   2 OR MORE, RESYNC
0198   A 00EC 2F06                     BLE     RESYN3
0199   A 00EE 0CCD                     INC     BSYNCT   LESS THAN 2:
0200   A 00F0 0A1D                     DEC     HDRCT    JUST UPDATE COUNTERS
0201   A 00F2 2033                     BRA     OUT
0202   A 00F4 8602           RESYN3    LDA     #02      SET SYNC MODE =2
0203   A 00F6 971B                     STA     SYMODT
0204   A 00F8 8601                     LDA     #01      SET STATUS BIT=1
0205   A 00FA 9A17                     ORA     STATUS
0206   A 00FC 9717                     STA     STATUS
0207   A 00FE 2027                     BRA     OUT
0208   A 0100 B61801         HDRCT4    LDA     RCVDAT   FIRST SYNC?
0209   A 0103 817E                     CMPA    #$7E
```

```
AMERICAN AUTOMATION ASSEMBLER          LINT    20  Dec  83          PAGE 006

0210   A 0105 260A                     BNE     BSYN4    NO, BRANCH
0211   A 0107 0FCD                     CLR     FSYNOK   YES, UPDATE COUNTERS
0212   A 0109 0A1D                     DEC     HDRCT
0213   A 010B 201A                     BRA     OUT
0214   A 010D 0CCD             BSYN4   INC     FSYNOK   FLAG BAD SYNC
0215   A 010F 0CD3                     INC     LCOMER   INCREMENT COMMUNICATION ERRORS
0216   A 0111 8602                     LDA     #02      MORE THAN 2 BAD SYNCS?
0217   A 0113 91CD                     CMPA    BSYNCT
0218   A 0115 2F06                     BLE     RESYN4   YES, RESYNC
0219   A 0117 0CCD                     INC     BSYNCT   NO,UPDATE COUNTERS
0220   A 0119 0A1D                     DEC     HDRCT
0221   A 011B 200A                     BRA     OUT
0222   A 011D 8602             RESYN4  LDA     #02      SET SYNC MODE =2
0223   A 011F 971B                     STA     SYMDCT
0224   A 0121 8601                     LDA     #01      SET STATUS BIT
0225   A 0123 9A17                     ORA     STATUS
0226   A 0125 9717                     STA     STATUS
0227   A 0127 1CAF             OUT     ANDC    $AF      ENABLE INTERRUPTS
0228   A 0129 3B                       RTI
0229                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER          LPMON   20 Dec 83          PAGE 001

0001                        ********************************************************
0002                        *                                                      *
0003                        * LPMON1 SUBROUTINE                                    *
0004                        *                                                      *
0005                        *   FUNCTION: PULSES AUTO-RESET CIRCUIT TO PREVENT     *
0006                        *             RESET, MONITORS AND CONTROLS RECEIVED    *
0007                        *             DATA LINES.                              *
0008                        *   INPUT:    FROUT: WHEN = 0 INDICATES A GOOD DATA    *
0009                        *                    FRAME HAS BEEN RECEIVED SINCE     *
0010                        *                    THE LAST TIME LPMON WAS EXECUTED. *
0011                        *                    IT SERVES AS A COUNTER TO COUNT   *
0012                        *                    PROGRAM LOOPS SINCE A GOOD FRAME. *
0013                        *   OUTPUT:   CHANNEL STATUS WORDS: CH0GD-CH7GD,       *
0014                        *                                   CH10GD-CH15GD.     *
0015                        *             STATUS BIT 4 UPDATE (BAD CHAN. DETECTION).*
0016                        *   USES:     A                                        *
0017                        *   CALLS:    NONE                                     *
0018                        *   CALLED BY MAIN                                     *
0019                        *                                                      *
0020                        ********************************************************
0021                        *
0022                        *
0023                        ****************SYMBOL DEFINITION**********************
0024                        *
0025                        *FROUT COUNTS MAIN PROGRAM LOOPS WITHOUT A GOOD FRAME
0026  A 0000 000000D9       FROUT   EQU     $D9
0027                        *
0028                        *STATUS IS THE LOCAL STATUS WORD
0029  A 0000 00000017       STATUS  EQU     $17
0030                        *
0031                        *CHANI IS A MEMORY IMAGE OF THE SELECTED LINK CHANNEL
0032  A 0000 000000DB       CHANI   EQU     $DB
0033                        *
0034                        *INSEL IS THE LINK INPUT CHAN. SELECT FLOP
0035  A 0000 00002000       INSEL   EQU     $2000
0036                        *
0037                        *RSTDIS IS THE AUTO RESET ADDRESS
0038  A 0000 00001000       RSTDIS  EQU     $1000
0039                        *
0040                        *CH0GD IS THE CHANNEL 0 GOOD FLAG
0041  A 0000 000000E1       CH0GD   EQU     $E1
```

```
AMERICAN AUTOMATION ASSEMBLER          LPMON   20  Dec  83          PAGE 002

0042                                *
0043                                *CH0TST IS THE CHANNEL 0 TESTED FLAG
0044   A 0000 000000E2              CH0TST   EQU     $E2
0045                                *
0046                                *CH1GD IS THE CHAN 1 GOOD FLAG
0047   A 0000 000000E3              CH1GD    EQU     $E3
0048                                *
0049                                *CH1TST IS THE CHANNEL 1 TESTED FLAG
0050   A 0000 000000E4              CH1TST   EQU     $E4
0051                                *
0052                                ***************START OF PROGRAM*********************
0053                                *
0054   A 0000 B71000                        STA     RSTDIS   PULSE AUTO RESET
0055   A 0003 0DD9                          TST     FROUT    FROUT = 0 ?
0056   A 0005 2705                          BEQ     FROUT0   YES: BRANCH
0057   A 0007 2D48                          BLT     FROUTN   NO: FROUT >=128 ?
0058                                *                         YES: BRANCH
0059   A 0009 0CD9                          INC     FROUT    NO: COUNT
0060   A 000B 39                            RTS     RETURN
0061                                *
0062   A 000C 0DDB                  FROUT0  TST     CHANI    CHAN. 0 ON ?
0063   A 000E 2729                          BEQ     CH0GN1   YES:BRANCH
0064   A 0010 4F                            CLRA    NO:      SET CH1 GOOD, TESTED
0065   A 0011 4C                            INCA
0066   A 0012 97E3                          STA     CH1GD
0067   A 0014 97E4                          STA     CH1TST
0068   A 0016 0DE2                          TST     CH0TST   CHAN 0 TESTED ?
0069   A 0018 2609                          BNE     CH0T1    YES: BRANCH
0070   A 001A 4F                            CLRA    NO:      TEST CH 0
0071   A 001B 97DB                          STA     CHANI
0072   A 001D B72000                        STA     INSEL
0073   A 0020 0CD9                          INC     FROUT
0074   A 0022 39                            RTS
0075                                *
0076   A 0023 0DE1                  CH0T1   TST     CH0GD    CHANNEL 0 GOOD ?
0077   A 0025 2609                          BNE     CH0G1    YES: BRANCH
0078   A 0027 8610                  BOUT1   LDA     #$10     NO: SET BAD CHAN STATUS
0079   A 0029 9A17                          ORA     STATUS
0080   A 002B 9717                          STA     STATUS
0081   A 002D 0CD9                          INC     FROUT    COUNT
0082   A 002F 39                            RTS
0083                                *
```

```
AMERICAN AUTOMATION ASSEMBLER          LPMON   20  Dec  83          PAGE 003

0084   A 0030 86EF          CH0G1    LDA     #$EF     RESET BAD CHAN STATUS
0085   A 0032 9417                   ANDA    STATUS
0086   A 0034 9717                   STA     STATUS
0087   A 0036 0CD9                   INC     FROUT    COUNT
0088   A 0038 39                     RTS
0089                        *
0090   A 0039 4F            CH0ON1   CLRA    SET      CH 0 GOOD; TESTED
0091   A 003A 4C                     INCA
0092   A 003B 97E1                   STA     CH0GD
0093   A 003D 97E2                   STA     Ch0TST
0094   A 003F 0DE4                   TST     CH1TST   CHAN. 1 TESTED?
0095   A 0041 2608                   BNE     CH1T1    YES: BRANCH
0096   A 0043 B72000                 STA     INSEL    NO: TEST CHAN 1
0097   A 0046 97DB                   STA     CHAN1
0098   A 0048 0CD9                   INC     FROUT
0099   A 004A 39                     RTS
0100                        *
0101   A 004B 0DE3          CH1T1    TST     CH1GD    CHAN 1 GOOD?
0102   A 004D 26E1                   BNE     CH0G1    YES: BRANCH
0103   A 004F 20D6                   BRA     BOUT1    NO: SET BAD CHAN STATUS BIT
0104                        *
0105   A 0051 0DDB          FROUTN   TST     CHAN1    CHAN 0 ON ?
0106   A 0053 272A                   BEQ     CH0ON2   YES: BRANCH
0107                        *                  NO:
0108   A 0055 0DE2                   TST     CH0TST   CHAN 0 TESTED ?
0109   A 0057 260A                   BNE     CH0T2    YES: BRANCH
0110   A 0059 4F                     CLRA    NO:      SWITCH TO CH 0
0111   A 005A 97DB                   STA     CHAN1
0112   A 005C B72000                 STA     INSEL
0113   A 005F 4C                     INCA
0114   A 0060 9709                   STA     FROUT    FROUT = 1
0115   A 0062 39                     RTS
0116                        *
0117   A 0063 0DE1          CH0T2    TST     CH0GD    CHAN 0 GOOD ?
0118   A 0065 2619                   BNE     CH0G2    YES: BRANCH
0119   A 0067 4F                     CLRA    NO:      SWITCH TO CHAN 0
0120   A 0068 97DB                   STA     CHAN1
0121   A 006A B72000                 STA     INSEL
0122   A 006D 97E1          BOUT2    STA     CH0GD    RESET ALL FLAGS
0123   A 006F 97E2                   STA     CH0TST
0124   A 0071 97E3                   STA     CH1GD
0125   A 0073 97E4                   STA     CH1TST
```

```
AMERICAN AUTOMATION ASSEMBLER          LPMON   20  Dec  83          PAGE 004

0126   A 0075 8601                     LDA     #01       SET FROUT=1
0127   A 0077 97D9                     STA     FROUT
0128   A 0079 8611                     LDA     #$11      SET BAD CHAN AND LOOKING
0129   A 007B 9A17                     ORA     STATUS    FOR SYNC STATUS BITS
0130   A 007D 9717                     STA     STATUS
0131   A 007F 39                       RTS
0132                          *
0133   A 0080 4F              CH0G2    CLRA
0134   A 0081 97DB                     STA     CHAN1     SWITCH TO CHAN 0
0135   A 0083 B72000                   STA     INSEL
0136   A 0086 97D9                     STA     FROUT     SET UP FOR FROUT=1
0137   A 0088 97E3                     STA     CH1GD     INDICATE CH 1 BAD
0138   A 008A 4C                       INCA    INDICATE CH 1 BAD
0139   A 008B 97E4                     STA     CH1TST
0140   A 008D 2098                     BRA     BOUT1     SET BAD CHAN STATUS
0141                          *
0142   A 008F 0DE4            CH0ON2   TST     CH1TST    CH 1 TESTED ?
0143   A 0091 260A                     BNE     CH1T2     YES: BRANCH
0144   A 0093 4F                       CLRA    NO:       SWITCH TO CH 1
0145   A 0094 4C                       INCA
0146   A 0095 97DB                     STA     CHAN1
0147   A 0097 B72000                   STA     INSEL
0148   A 009A 97D9                     STA     FROUT     FROUT= 1
0149   A 009C 39                       RTS
0150                          *
0151   A 009D 0DE3            CH1T2    TST     CH1GD     CHANNEL 1 GOOD ?
0152   A 009F 260A                     BNE     CH1G2     YES: BRANCH
0153   A 00A1 4F                       CLRA    NO:SWITCHTO CH 1
0154   A 00A2 4C                       INCA
0155   A 00A3 97DB                     STA     CHAN1
0156   A 00A5 B72000                   STA     INSEL
0157   A 00A8 4F                       CLRA    GO        RESET ALL FLAGS
0158   A 00A9 20C2                     BRA     BOUT2     AND SET BAD CHAN STATUS
0159                          *
0160   A 00AB 4F              CH1G2    CLRA    SET       UP FOR FROUT=1
0161   A 00AC 97D9                     STA     FROUT
0162   A 00AE 97E1                     STA     CH0GD     INDICATE CH 0 BAD
0163   A 00B0 4C                       INCA
0164   A 00B1 97E2                     STA     CH0TST    SET CH 0 BAD
0165   A 00B3 97DB                     STA     CHAN1     SWITCH TO CHAN 1
0166   A 00B5 B72000                   STA     INSEL
0167   A 00B8 16FF6C                   LBRA    BOUT1     SET BAD CHAN STATUS
0168                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER          LXMIT   20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * LXMIT SUBROUTINE                                         *
0004                        *                                                          *
0005                        *   FUNCTION: PUTS A BYTE IN THE LINK UART DATA BUF.       *
0006                        *             TIMES OUT IF UART IS NOT READY SOON          *
0007                        *             ENOUGH.   UPDATES STATUS BIT 3.              *
0008                        *   INPUT:    BYTE IN A REG.                               *
0009                        *   OUTPUT:   BYTE INTO LINK UART DATA BUF, IF NO          *
0010                        *             TIME-OUT; UPDATED STATUS BIT                 *
0011                        *   USES:     B,X                                          *
0012                        *                                                          *
0013                        ************************************************************
0014                        *
0015                        *
0016                        *STATUS IS USED TO KEEP LOCAL STATUS INFO
0017  A 0000 00000017       STATUS  EQU     $17
0018                        *
0019                        *LUCON IS LINK UART CONTROL REG.
0020  A 0000 00001800       LUCON   EQU     $1800
0021                        *
0022                        *LUDAT IS LINK UART DATA REG.
0023  A 0000 00001801       LUDAT   EQU     $1801
0024                        *
0025                        *
0026                        ********************START OF PROGRAM*********************
0027                        *
0028                        *
0029  A 0000 8E000A         LXMIT   LDX     #$000A  SET UP TIME-OUT CTR.
0030  A 0003 C602                   LDB     #02     TEST LUART READY
0031  A 0005 F51800         XWAIT   BITB    LUCON
0032  A 0008 260B                   BNE     SEND
0033  A 000A 301F                   LEAX    -1,X    BRANCH IF NOT READY
0034  A 000C 26F7                   BNE     XWAIT
0035  A 000E C608           XTOUT   LDB     #08     TIME-OUT STATUS UPDATE
0036  A 0010 DA17                   ORB     STATUS
0037  A 0012 D717                   STB     STATUS
0038  A 0014 39                     RTS
0039  A 0015 B71801         SEND    STA     LUDAT   SEND DATA
0040  A 0018 C6F7                   LDB     #$F7    UPDATE STATUS
0041  A 001A D417                   ANDB    STATUS
0042  A 001C D717                   STB     STATUS
0043  A 001E 39                     RTS
0044                                END
```

```
AMERICAN AUTOMATION ASSEMBLER          LTSTER  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * LTSTER SUBROUTINE                                        *
0004                        *                                                          *
0005                        *   FUNCTION: TESTS ALTERNATE LINK CHANNEL. IF DATA IS *
0006                        *             RECEIVED, AN SWI TO LINT SUBROUTINE IS   *
0007                        *             EXECUTED TO PROCESS IT                       *
0008                        *   INPUT:    CHAN1, CONTAINING INPUT CHANNEL ACTIVE   *
0009                        *   OUTPUT:   CHANGES CHAN1; GIVES DATA BYTE TO LINT IF*
0010                        *             ONE IS RECEIVED. INDICATES TIME-OUT TO   *
0011                        *             CALLING PROGRAM WITH ZERO IN X-REG.      *
0012                        *   USES:     A,X                                        *
0013                        *   CALLED BY LPMON                                      *
0014                        *   CALLS:                                               *
0015                        *                                                          *
0016                        ************************************************************
0017                        *
0018                        *
0019                        ******************SYMBOL DEFINITION**************************
0020                        *
0021                        *LINTEN ENABLES RECEIVE INTERRUPTS, 8 BIT, NO PARITY
0022                        * 1 STOP BIT OPERATION OF ACIA
0023  A 0000 00000095       LINTEN  EQU     $95
0024                        *
0025                        *LINCON IS LINK ACIA CONTROL REG
0026  A 0000 00001800       LINCON  EQU     $1800
0027                        *
0028                        *CHAN1 IS MEMORY IMAGE OF CHANNEL SELECT BIT
0029  A 0000 000000DB       CHAN1   EQU     $DB
0030                        *
0031                        *INSEL IS ADDRESS OF LINK CHAN. SEL. FLOP
0032  A 0000 00002000       INSEL   EQU     $2000
0033                        *
0034                        *WAIT IS A 30 MILLISECOND WAIT COUNT
0035  A 0000 00000753       WAIT    EQU     $753
0036                        *
0037                        ******************START OF PROGRAM************************
0038                        *
0039  A 0000 3617           LTSTER  LDA     #$17    DISABLE LINK INTERRUPTS
0040  A 0002 B71800                 STA     LINCON
0041  A 0005 4F                     CLRA    SWITCH  CHANNEL
```

```
AMERICAN AUTOMATION ASSEMBLER          LTSTER  20  Dec  83          PAGE 002

0042   A 0006 0DDB                     TST     CHANI
0043   A 0008 2601                     BNE     CONT1
0044   A 000A 4C                       INCA
0045   A 000B 97DB           CONT1     STA     CHANI     STORE CHAN. IMAGE IN MEM.
0046   A 000D B72000                   STA     INSEL     SWITCH CHAN SEL FLOP
0047   A 0010 8601                     LDA     #01       SET UP TEST
0048   A 0012 8E0753                   LDX     #WAIT
0049   A 0015 301F           LOOP1     LEAX    -1,X      CHECK TIME OUT
0050   A 0017 2706                     BEQ     TOUT
0051   A 0019 B51800                   BITA    LINCON    DATA RECEIVED?
0052   A 001C 27F7                     BEQ     LOOP1     IF NO, LOOP UNTIL TIME OUT
0053   A 001E 3F                       SWI     IF        YES GO PROCESS
0054   A 001F 8695           TOUT      LDA     #LINTEN   ENABLE INTERRUPT IN LINK ACIA
0055   A 0021 B71800                   STA     LINCON
0056   A 0024 39                       RTS
0057                                   END
```

```
AMERICAN AUTOMATION ASSEMBLER          MAIN    20  Dec  83          PAGE 001

0001                        ********************************************************
0002                        *                                                      *
0003                        * MAIN ROUTINE                                         *
0004                        *                                                      *
0005                        *   FUNCTION:CONTROLS PROGRAM FLOW                     *
0006                        *   INPUT:   NONE                                      *
0007                        *   OUTPUT:  NONE                                      *
0008                        *   USES:    NONE                                      *
0009                        *   CALLS:   LPMON,SYNCTR,ANACTR,DIGINT,ENDTR,DISPLY.  *
0010                        *   CALLED BY NONE (ENTRY FROM INIT ROUTINE)           *
0011                        *                                                      *
0012                        ********************************************************
0013                        *
0014                        *
0015                        ****************SYMBOL DEFINITION**********************
0016                        *
0017                        *LPMON IS A SUBROUTINE WHICH MONITORS VARIOUS PARAMETRS
0018                        *   ONCE THRU EACH MAIN PROGRAM LOOP
0019  A 0000 0000FE60       LPMON    EQU     $FE60
0020                        *
0021                        *SYNCTR IS A SUBROUTINE WHICH TRANSMITS THE SYNC WORDS
0022  A 0000 0000FB30       SYNCTR   EQU     $FB30
0023                        *
0024                        *ANACTR IS A SUBROUTINE WHICH TRANSMITS STATUS NUMBER
0025                        *   OF BYTES AND ANALOG VALUES, AS WELL AS CONTROLS
0026                        *   AND MONITORS ANALOG ACQUISITION
0027  A 0000 0000F500       ANACTR   EQU     $F500
0028                        *
0029                        *DIGINT IS A SUBROUTINE WHICH ACQUIRES AND TRANSMITS
0030                        *   THE DIGITAL INPUTS
0031  A 0000 0000FB40       DIGINT   EQU     $FB40
0032                        *
0033                        *ENDTR IS A SUBROUTINE WHICH TRANSMITS ALL THE STUFF
0034                        *   AT THE END OF THE DATA FRAME
0035  A 0000 0000FB50       ENDTR    EQU     $FB50
0036                        *
0037                        *DISPLY IS A SUBROUTINE WHICH UPDATES FRONT PANEL AND
0038                        *   USER MONITOR DISPLAYS
0039  A 0000 0000FCE0       DISPLY   EQU     $FCE0
0040                        *
0041                        *UPDATE IS A SUBROUTINE WHICH UPDATES USER OUTPUTS WHEN
```

```
AMERICAN AUTOMATION ASSEMBLER          MAIN    20 Dec 88          PAGE 002

0042                        *   A VALID FRAME IS RECEIVED
0043   A 0000 0000FDF0      UPDATE  EQU     $FDF0
0044                        *
0045                        ******************START OF PROGRAM************************
0046                        *
0047   A 0000 BDFE60        MAIN    JSR     LPMON
0048   A 0003 BDFB30                JSR     SYNCTR
0049   A 0006 BDF500                JSR     ANACTR
0050   A 0009 BDFB40                JSR     DICTR
0051   A 000C BDFB50                JSR     ENDTR
0052   A 000F BDFDE0                JSR     DISPLY
0053   A 0012 20EC                  BRA     MAIN
0054                                END
```

```
AMERICAN AUTOMATION ASSEMBLER          SYNCTR  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * SYNCTR SUBROUTINE                                        *
0004                        *                                                          *
0005                        *   FUNCTION:TRANSMITS THE LINK SYNC BYTES                 *
0006                        *   INPUT:   NONE                                          *
0007                        *   OUTPUT:  SYNC WORDS TO LINK UART                       *
0008                        *   USES:    A,LXMIT                                       *
0009                        *                                                          *
0010                        ************************************************************
0011                        *
0012                        *
0013                        ******************SYMBOL DEFINITIONS**********************
0014                        *
0015                        *SYNCWD IS THE LINK SYNC WORD
0016  A 0000 0000007E       SYNCWD   EQU     $7E
0017                        *
0018                        *LXMIT IS THE LINK HANDLER SUBROUTINE
0019  A 0000 0000FB00       LXMIT    EQU     $FB00
0020                        *
0021                        *
0022                        ******************START OF PROGRAM************************
0023                        *
0024  A 0000 867E           SYNCTR   LDA     #SYNCWD
0025  A 0002 BDFB00                  JSR     LXMIT
0026  A 0005 BDFB00                  JSR     LXMIT    LOAD AND TRANSMIT SYNC WORD
0027  A 0008 39                      RTS
0028                                END
```

```
AMERICAN AUTOMATION ASSEMBLER          UACINT  20  Dec  83          PAGE 001

0001                        ************************************************************
0002                        *                                                          *
0003                        * UACINT SUBROUTINE                                        *
0004                        *                                                          *
0005                        *   FUNCTION:HANDLES USER ACIA INTERRUPTS                  *
0006                        *   INPUT:   FROM USER ACIA                                *
0007                        *   OUTPUT:  NONE                                          *
0008                        *   USES :   A                                             *
0009                        *   CALLS:   NONE                                          *
0010                        *   CALLED BY NONE (ENTRY FROM USER ACIA INTERRUPT)        *
0011                        *                                                          *
0012                        ************************************************************
0013                        *
0014                        *
0015                        ******************SYMBOL DEFINITION*************************
0016                        *
0017                        *UACRB IS THE USER ACIA READ BUFFER
0018  A 0000 00002801       UACRB    EQU     $2801
0019                        *
0020                        *
0021                        ******************START OF PROGRAM**************************
0022                        *
0023                        *READ USER INTERFACE TO CLEAR INTERRUPT
0024  A 0000 B62801         UACINT   LDA     UACRB
0025  A 0003 3B                      RTI
0026                                 END
```

```
AMERICAN AUTOMATION ASSEMBLER          UPDATE  20  Dec  83          PAGE 001

0001                    ************************************************************
0002                    *                                                          *
0003                    * UPDATE SUBROUTINE                                        *
0004                    *                                                          *
0005                    *   FUNCTION: UPDATES USER ANALOG AND BINARY OUTPUTS       *
0006                    *   INPUTS:   DATA IN A RECEIVED DATA BUFFER               *
0007                    *             FRRDY FLAG TO SIGNAL DATA AVAILABLE          *
0008                    *             DSWS TO SELECT BETWEEN DIGITAL OUTPUTS       *
0009                    *               16 TO 31 AND LIMIT BITS 0 TO 15            *
0010                    *   OUTPUTS:  31 ANALOG SIGNALS                            *
0011                    *             DIGITAL OUTPUTS 0 TO 15                      *
0012                    *             DIGITAL OUTPUTS 16 TO 31 OR ANALOG LIMIT     *
0013                    *               BITS 0 TO 15                               *
0014                    *   USES:     A,X,Y,U                                      *
0015                    *                                                          *
0016                    ************************************************************
0017                    *
0018                    *
0019                    ******************SYMBOL DEFINITION**************************
0020                    *
0021                    *FRRDY IS USED TO FLAG A COMPLETE READY FRAME
0022  A 0000 000000C7   FRRDY   EQU    $C7
0023                    *
0024                    *RCVDGD FLAGS THAT GOOD DATA HAS BEEN RECEIVED
0025  A 0000 000000D7   RCVDGD  EQU    $D7
0026                    *
0027                    *RBUF0 IS THE START ADDR. OF A RECEIVED DATA TABLE
0028  A 0000 00000020   RBUF0   EQU    $20
0029                    *
0030                    *RBUF1 IS THE START ADDR. OF A RECEIVED DATA TABLE
0031  A 0000 00000070   RBUF1   EQU    $70
0032                    *
0033                    *RBUFSE POINTS TO THE READY DATA BUFFER
0034  A 0000 000000C1   RBUFSE  EQU    $C1
0035                    *
0036                    *AOUT0 IS THE START ADDR. OF THE FIRST ANALOG OUT CARD
0037  A 0000 00005000   AOUT0   EQU    $5000
0038                    *
0039                    *AOUT1 IS THE START ADDR. OF THE SECOND ANALOG OUT CARD
0040  A 0000 00005800   AOUT1   EQU    $5800
0041                    *
```

```
AMERICAN AUTOMATION ASSEMBLER          UPDATE  20  Dec  83          PAGE 002

0042                                *AOUT2 IS THE START ADDR. OF THE THIRD ANALOG OUT CARD
0043   A 0000 00006000              AOUT2    EQU     $6000
0044                                *
0045                                *AOUT IS THE START ADDR. OF THE SUBROUTINE WHICH LOADS
0046                                * THE ANALOG OUTPUT LATCHES
0047   A 0000 0000FDE0              AOUT     EQU     $FDE0
0048                                *
0049                                *DOUTL IS THE ADDR. OF THE LATCHES FOR DOUT 0 - 15
0050   A 0000 00005012              DOUTL    EQU     $5012
0051                                *
0052                                *DOUTH IS THE ADDR. OF THE LATCHES FOR DOUT 16 - 31
0053   A 0000 00005812              DOUTH    EQU     $5812
0054                                *
0055                                *DSW4 IS TEH ADDR. OF A USER CONTROL SWITCH
0056                                *  BIT 7 IS USED TO INDICATE LIMIT BIT OR DIGITAL OUTPUT
0057   A 0000 0000A000              DSW4     EQU     $A000
0058                                *
0059                                *FROUT IS USED TO COUNT MAIN PROG. LOOPS WITHOUT
0060                                *A GOOD FRAME OUTPUT . IT IS RESET BY THIS ROUTINE
0061                                *IF THIS ROUTINE OUTPUTS A NEW FRAME
0062   A 0000 00000A09              FROUT    EQU     $D9
0063                                *
0064                                *DOFF IS THE OFFSET OF THE DIGITAL WORDS IN THE MEMORY
0065                                * DAT BUFFER.
0066   A 0000 00000030              DOFF     EQU     $30
0067                                *
0068                                *
0069                                ******************START OF PROGRAM***********************
0070                                *
0071   A 0000 0DD7          UPDATE  TST     RCVDGD  INC RCVDGD IF FIRST FRAME
0072   A 0002 2602                  BNE     CONT2
0073   A 0004 0CD7                  INC     RCVDGD
0074   A 0006 108E5000      CONT2   LDY     #AOUT0  OUTPUT TO ANALOG CARD 0
0075   A 000A 8609                  LDA     #09     TELL HOW MANY WORDS TO OUTPUT
0076   A 000C BDFDF0                JSR     AOUT
0077   A 000F 108E5800              LDY     #AOUT1  OUTPUT TO ANALOG CARD 1
0078   A 0013 8609                  LDA     #09
0079   A 0015 BDFDE0                JSR     AOUT
0080   A 0018 108E6000              LDY     #AOUT2  OUTPUT TO ANALOG CARD 2
0081   A 001C 8606                  LDA     #06
0082   A 001E BDFDE0                JSR     AOUT
0083   A 0021 EE81                  LDU     ,X++
0084   A 0023 FF5012                STU     DOUTL
0085   A 0026 B6A000                LDA     DSW4    WANT LIM BITS OR DOUT 15-31?
0086   A 0029 2D02                  BLT     CONT3   TEST DSW4 TO FIND OUT
0087   A 002B 3002                  LEAX    2,X     POINT TO LIM BITS
0088   A 002D EE84          CONT3   LDU     0,,     OUTPUT ONE OR THE OTHER
0089   A 002F FF5812                STU     DOUTH
0090   A 0032 0FD9                  CLR     FROUT   INDICATE A GOOD FRAME RCV'D
0091   A 0034 39                    RTS
0092                                END
```

6.0 How to Put in the Right Checksum

This section assumes the user is familiar with the use of a PROM programmer, and with hexadecimal notation. Access to a hexadecimal calculator such as Dwayne Schiebel's may be helpful.

Three locations in each ROM are saved for storing the ROM checksum and checksum adjust. Let A be the location for checksum adjust, B be the location for checksum MSB, and C be the location for the LSB.

Values to plug into A, B, and C are found as follows:

1. Get the ROM program into the programmer's RAM.
2. Set locations A, B, and C to zero (0080-0082 in 10A ROM, 07FD-07FF in 6A ROM).
3. Do a checksum on 0000 to 07FF.
4. Let the resulting two LSD's be called Z, and the next two LSD's be called Y; e.g., if the programmer gives 0467F2 as a checksum, Z = F2 and Y = 67.
5. If Y + Z = 0, then A = B = C = 00.

   If $0 < Y + Z \leq 100_H$, then

   $B = Y + 1$

   $C = Z$

   $A = 100_H - (B + C)$

   else

   $B = Y + 2$

   $C = Z$

   $A = 200 - (B + C)$

(All arithmetic done in hexadecimal.)

7.0 How to Enter Analog Limits

This section assumes the user is familiar with the use of a PROM programmer, and with hexadecimal notation. Access to a hexadecimal calculator such as Dwayne Schiebel's may be useful.

Minimum and maximum limits can be programmed into ROM 10A for each analog input. Input voltages that exceed these voltages cause a limit bit to be set. These limit bits, in turn, cause the front panel DVM to blink when the appropriate channel is selected; the limit bits for channels 0 to 15 can also be passed on to the user, instead of digital outputs 16 to 31.

These limits are programmed as follows. First, the appropriate value for the limit must be calculated. Assume it is V volts, where V is a decimal number between -10 and +9.996 volts. Calculate

$$N_D = ((V + 10)/20) \times 4096$$

Then change N_D to a hexidecimal number, N_H and round, if necessary, to an integer. (Dwayne Schiebel and others have calculators that will make this conversion at the touch of a button.) Let XYZ be the three character hex representation of N_H. It is programmed into two consecutive bytes of ROM as follows:

Byte #	Contents
N	Ø X
N + 1	Y Z

The byte to min/max channel mapping is specified in Table 5.

Once the min/max values are known they can be programmed by reading ROM 10A V1.0-RJL into the ROM programmer and changing the appropriate bytes. Then the checksum should be changed,

as described in Section 6.0. Finally, a new ROM can be programmed, or the old one erased and reprogrammed. A new and different label should be used to identify it.

TABLE 5

Analog Voltage Limit Locations in Rom 10A

Byte #	Channel	Min	Max
0, 1	0		X
2, 3	0	X	
4, 5	1		X
6, 7	1	X	
8, 9	2		X
A, B	2	X	
C, D	3		X
F, F	3	X	
10, 11	4		X
. . .	. . .	. . .	
7C, 7D	31		X
7E, 7F	31	X	

7.1 Example

Assume we want to change the default min and max values from the V1.0-RJL standard to -5 and +5 volts, respectively for channel 17. Calculate

$$((-5 + 10)/20) \times 4096 = 1024_{10} = 400_{16}$$

$$((+5 + 10)/20) \times 4096 = 3072_{10} = C00_{16}$$

Interpolating from Table 4, channel 17 max corresponds to bytes 44_{16} and 45_{16} and channel 17 min to bytes 46_{16} to 47_{16}.

Read ROM 10A V1.0-RJL into the PROM Programmer. Change the contents of the above-mentioned four locations as follows:

Byte # (Hex)	Contents (Hex)
44	0C
45	00
46	04
47	00

Find and enter the new checksum as described in Section 6.0 and program a new ROM.

8.0 Changing Default Power-On Digital Outputs

This section assumes the user is familiar with the use of a PROM programmer and with the hexadecimal number system.

The FEDAL Digital Outputs go to specified default values shortly after power-on. In V1.0-RJL firmware the default is all logic high. (Of course, after a good data frame is received these outputs are updated.) It is not too hard to conceive of applications in which certain switch settings are desirable immediately after power-on. This can be effected by modifying these default values.

The defaults are modified as follows. Table 6 specifies the default values table located in ROM 10A. Each output is assigned one bit in this table, and changing a bit is as easy as reprogramming the byte in which it resides, and then updating the checksum as described in Section 6.0.

8.1 Example

Suppose we want outputs 6, 12, and 30 to default to logic low and all the rest to logic high. We need to program

$10111111_2 = BF_{16}$ into location 88

$11101111_2 = FF_{16}$ into location 89

$10111111_2 = BF_{16}$ into location 8B

of ROM 10A, and then update the ROM checksum.

TABLE 6

Default Digital Output Locations in ROM 10A.

Byte # (Hex)	Default Output # (decimal) (MSB .....LSB)							
88	7	6	5	4	3	2	1	0
89	15	14	13	12	11	10	9	8
8A	23	22	21	20	19	18	17	16
8B	31	30	29	28	27	26	25	24

9.0 Test Procedures

This section includes test procedures for the three Front-End Data Link (FEDAL) boards and the chassis. Card schematics, parts lists, and layouts, however, are not included. These can be found on pages 26 to 62 of the FEDAL report, EDIR No. 239.

TEST PROCEDURES

FEDAL-In Card

FEDAL-MP Card

FEDAL Output Card Tests

FEDAL Chassis

TEST PROCEDURE FOR FEDAL-IN CARD

R. Lacasse

June 10, 1983

1. Remove all cards and power supply connections from the front end data tester.

2. Put the blank, wired, FEDAL-IN card in the input slot.

3. Test for shorts between the tester's supply inputs.

4. Verify continuity between:

   A. +15 V supply input and: B34, C52, G17, G33, H16, H18, H24, J18, J30, J33, Z48, Z55.

   B. 15 V RET input and: B38, B39, G12, G13, G28, G44, H19, H20, H26, J19, J20, L15, M12, M15, M17.

   C. -15 V supply input and: B43, B53, E18, E34, H08, H21, H27, H29, H32, J21, Y48, Y55.

   D. +5 V supply input and: B01, B12, B23, D01, D09, F02, H12, H13, H14, L07, L09.

   E. 5 V RTN supply and: C10, C21, C32, E07, E15, F12, F13, G09, L10.

5. Remove the card and install all its components per the layout on page 4.

6. Connect supplies to the tester.

7. Turn on the supplies and measure the currents consumed by the tester with no cards inserted.

8. Plug in the card and measure the incremental currents, consumed by the card. The currents should be in the following ranges:

	Minimum (mA)	Maximum (mA)
+5 V	103	245
+15 V	44	72
-15 V	44	72

9. Install the "Test 6" ROM into location 10A of the FEDAL-MP board. Install the FEDAL-MP board and FEDAL-IN cards into their proper slots in the front end data link chassis. Remove the top of the chassis for easy access to the FEDAL-IN board.

10. Install "clothes-pins" on both MUX-16's, locations 3B and 3C. Remove the 240 ohm resistor between H31 and J31. Install one end of a 1 KΩ, 1/8 W, 5% resistor into J31; leave the other end of the resistor sticking up.

11. Turn the power on and verify the presence of low-going 0.75 μsec pulses on 3A12, 3A13, 3A14, 3A15. Verify the presence of a TTL low on 2B06. Measure the width of the TTL-high part of the waveform at 2B01. It should be ≤ 25 μsec, typically, 20 μsec.

12. Trim the gain and offset of the ADC 80 as described in this section. The gain and offset pots are located as shown on the component layout, page 4. Use the voltage calibrator supply (Digitec, Model 311) to apply -9.9976 volts to channel zero of the MUX by connecting the positive supply output to 3B19 and the negative output to end of the 1 KΩ resistor that was left sticking up. (Use the "REVERSE" feature of the 311 to get the negative voltage.) The inverted ADC output is displayed on the FEDAL chassis' status display and the channel to be monitored is selected with the top digiswitch on the front panel. Select channel ØØ. Adjust the offset pot so that the display flickers between ØØØØ and ØØØ1. Apply +9.9927V with the Digitec and adjust the gain pot so that the display flickers between ØFFE and ØFFF. Double check both pot settings and readjust if necessary. Verify that an input of 0.0000 V gives a reading of Ø8ØØ ± 1.

13. Verify the channel select logic as follows:

    A. Select channel ØØ with the digiswitch.

    B. With no signal applied to 3B19, the display should not have a stable reading.

    C. Then apply +5.000 V from the Digitec to 3B19. The display should become stable and read ØCØØ ± 1.

    D. Repeat this test for channels 1-31. Corresponding MUX pins are shown in Table 1 on page 3.

14. Measure the inter-channel isolation of the MUXes as follows:

    A. Monitor channel zero with 3B19 connected to 3B12 with an EZ-hook. The display should read Ø8ØØ ± 1.

    B. Apply +10 V from the Digitec to 3B20. The reading should change by, at most, 1.

    C. Now apply -10 V. The reading should again change by, at most, 1. (Ideally, it should not change at all.)

    D. Repeat this test for the MUX at 3C: connect 3C19 to 3C12; apply ± 10 V to 3C20 and verify the display change as above, monitoring channel 16.

TABLE 1

0	3B19	12	3B07	24	3C11
1	20	13	6	25	10
2	21	14	5	26	9
3	22	15	4	27	8
4	23	16	3C19	38	7
5	24	17	20	29	6
6	25	18	21	30	5
7	26	19	22	31	4
8	11	20	23		
9	10	21	24		
10	9	22	25		
11	8	23	26		

15. Measure the slew rate of the INA-101 amplifier. Using a signal generator, apply a square wave, that goes from +10 V to -10 V with frequency of about 1 kHz, to 3B19. Ground of the signal generator should be on the end of the 1 KΩ test resistor that was left sticking up. Select channel ØØ with the digiswitch. Monitor PIN 14 on the ADC-80 with a scope. Verify that the settling time, including the ramp and ringing, on both transitions is less than 150 μsec.

16. Replace the 240 Ω resistor.

TEST PROCEDURE FOR FEDAL-MP CARD

R. J. Lacasse

June 23, 1983
Revised: 7/18/83 and 8/23/83

1. Remove all cards and power supply connections from the front-end data link tester (FEDALT).

2. Put the blank, wired, FEDAL-MP card in the μP slot. (If discrete components are already inserted, they may be left in, with the exception of 3G and 8F.)

3. Test for shorts between the FEDALT's supply inputs.

4. Verify continuity between:

+5 V and	B 1, 33, 47	5 V RTN and	B 2
	C 42, 14		C 1, 10, 15, 39, 45, 54
	D 1, 20, 24, 47		D 23, 27
	E 33, 38, 42		E 10, 36, 41, 45, 49, 51, 54
	F 17, 29, 32, 34, 47		F 30, 31
	G		G 23, 43, 54
	H 47		H 2, 3, 16, 17, 41, 36, 32
	J 38		J 41, 45, 49, 51, 54, 30, 34
	K 9, 11, 13, 23, 25, 27, 47		K 2, 16
	L 2, 5		L 54
	M 47		M
	N		N 13, 54
	P		P
	R		R
	S 1, 10, 12, 21, 33, 41		S 9
	T 1, 9, 12, 20, 51		T 7, 10, 21, 39, 53
	U 2, 10, 14, 20, 33, 41		U 13, 4, 22
	V 12, 16, 17		V 2, 9, 11, 15, 18, 27, 39, 47, 52, 53, 54, 55, 25
	W 2, 5, 44		W
	X		X
	Y 29		Y 13, 23, 55
	Z		Z

+15 V and B 14
J 29

15 V RTN and B 15, 18

-15 V and B 17
J 33

5. Remove the card and install all its components per the layout on page 5. The 2716 ROM at location 6A should be labeled FØØØ; the 2716 ROM at location 10A should be labeled TEST 7. Double check the component layout and orientation.

6. Connect supplies to the tester.

7. Turn on the supplies and measure the currents consumed by the tester with no cards inserted. Adjust the supply voltages if required to ± 0.1 V of nominal.

8. Current Consumption Test. Power off.

   Plug the card into the MP slot. Power on. Measure the incremental currents consumed by the card. These should be in the following ranges:

	Minimum (mA)	Typ. (mA)	Maximum (mA)
+5 V	600	800	1500
+15 V	5	7	15
-15 V	5	8	15

9. Power on Reset Test. Power off.

   Replace the 0.22 µF capacitor at 9B16 to 9B01 with a 15 µF capacitor, plus side to pin 16. Monitor 9A13 with a scope set to 0.5 sec per horizontal division and 2 V per vertical division. When +5 V power is turned on, a 1.0 ± 0.2 sec positive pulse should be seen on this pin. If not, reference schematic sheet 2. Replace the 0.22 µF capacitor.

10. User Baud Rate Generator Test.

    Verify a square wave with 52.1 ± 2 µsec period at 4B03 and 4B04. If not, see schematic sheet 10.

11. Twelve Volt Regulator Test

    Verify +12 V DC ± 1 V at 4K08 (75150), and -12 V DC ± 1 V at 4K05. If not, see schematic sheet 11.

12. Three Volt Regulator Test.

    Verify +3 V DC ± 0.5 V at 3G04 (100 Ω). If not, see schematic sheet 12.

13. 1 MHz Clock Test.

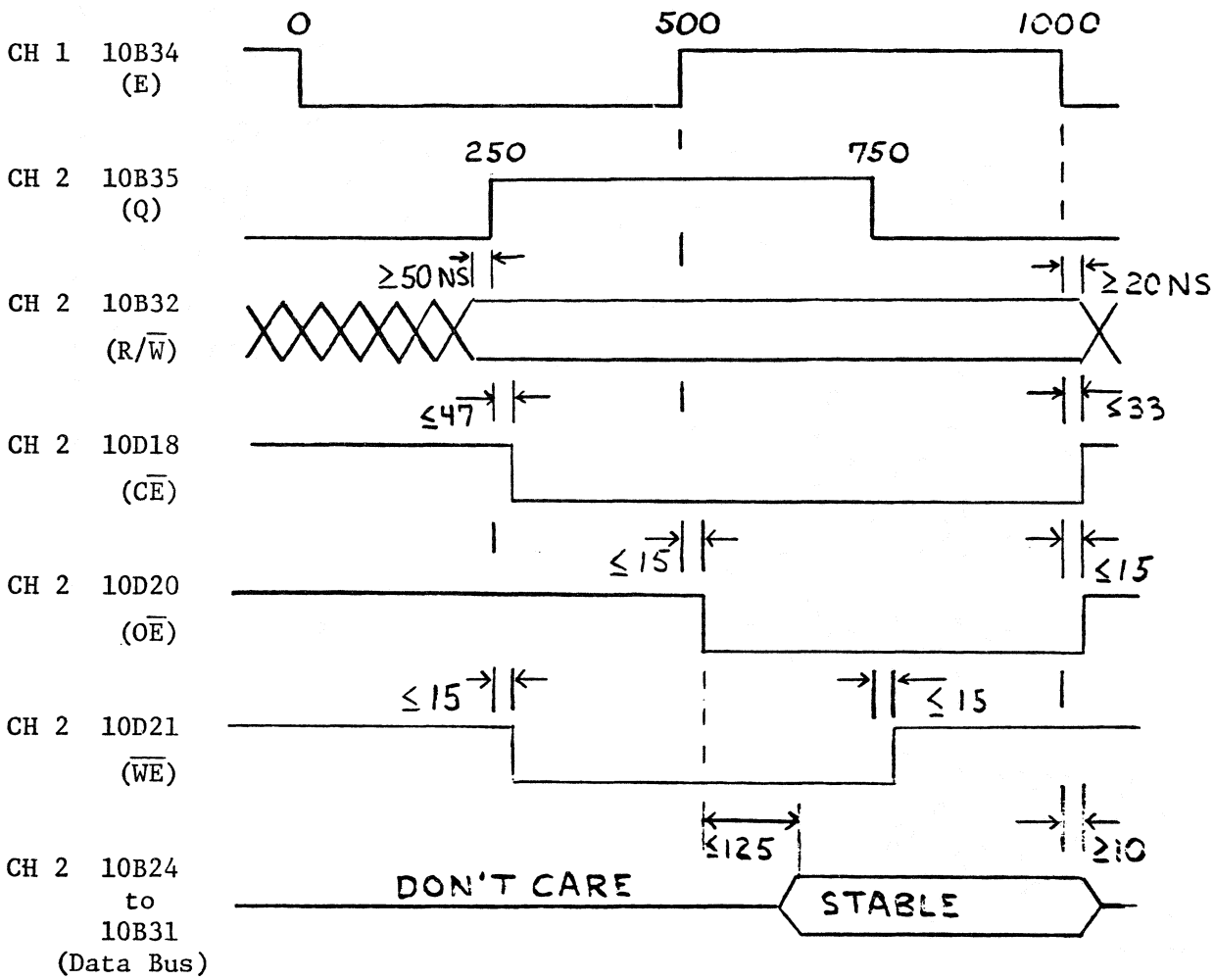
    Verify 1 MHz square wave at 10B34, 10B35. If not, see schematic sheet 1.

14. Signal Level and Timing Tests.

    Remove all boards from the front-end data link chassis (FEDALC). Insert the board under test into the "MP" slot. Place select switch on the front panel to status. Power on. Verify a TTL logic high on 10B 2, 3, 4, 37. Verify reasonable TTL logic switching signals on 10B 8 through 23.

14. (Continued):

Place the select switch in Restart Errors position. Verify the following timing. (Times in nanoseconds, always trigger on scope channel 1.)



Reference schematic sheets 1, 8, 9, if necessary.

15. Enable Signal Test.

Place the select switch in Status.

Verify a 0.75 ± .02 μsec pulse on 5F07. Trigger on this waveform. Use the second scope channel to verify similar pulses, each occurring about 19 μsec later than the previous one on:

5F 9, 10, 11, 12, 13, 14, 15
3F 7, 9, 10, 11, 12, 15
1F 7, 9, 10, 11, 12, 13, 14
2F 14, 13, 12

Reference schematic sheets 15 and 16 if necessary.

16. Digiswitch and Display Tests.

Verify that the value on the digiswitches associated with the front panel DVM is displayed on the "Remote" LED display. Try a few values and see that they are all displayed correctly. Do the same for the lower digiswitches, associated with the analog meter; they should be displayed on the "Local" LED display. If not, see schematic sheet 12.

17. Restart Error Counter Test.

Place LED display select switch in the restart error position. Turn power off. Replace the .47 µF capacitor at 9B09-9B08 with a 15 µF capacitor, with positive on capacitor to 9B09. Turn power on. The remote display should read 00. Turn power off. Replace the 200 K resistor at 9B07-9B10 with a 100 K resistor. Turn power on. The remote display should count cyclicly from ØØ to FF. If not, see schematic sheet 2. Replace the .47 µF capacitor and 200 KΩ resistor.

18. User UART Test.

Place LED display select switch in Communication Error position. Jumper pin 2 to pin 3 on the rear panel RS232 connector. Verify that the DVM digiswitches are displayed on the remote LED's. View the transmitted data on the jumpered pins. Verify a two level waveform, with the positive excursion $\geq$ 6 V and the negative excursion $\leq$ 6 V. If not, see schematic sheets 10, 11.

19. Link UART Tests. Turn power off.

Remove rear panel jumpers on the digital I/O Elco connector, if present. Install a FEDAL-OUT card on the bottom slot. Install the F800 ROM into 10A. Power on. Verify switching waveforms on the FEDAL-MP connector, pins 55, 56, 57, 58. Verify a roughly 1 Hz TTL switching waveform on 2E07 and 2F07 (6N137's). Verify that the remote status LED's are blanked. Power off. Reference schematic sheets 3 and 5 if required.

20. Link Receiver Tests.

Jumper A to C, B to D, E to J, and F to K on the rear panel digital I/O Elco connector. Power on. Verify a switching waveform on 2E06. Verify that both MV50's on the board light. Verify that with the select switch in any of the three positions, both remote and local displays read 00.

21. Auto-Reset Test.

Verify .75 µsec pulses with 20 $\pm$ 2 msec period on 9C01. Reference schematic sheet 2 if required.

TEST PROCEDURE FOR
FRONT-END DATA LINK OUTPUT CARD TESTS

R. Lacasse

August 23, 1983

1. If connected to the supplies, disconnect the front-end data link tester from the supplies.

2. Separate digital wires on rows G and V from analog wires on rows H and W. Pins 17 and 20 on the DACs are especially sensitive.

3. Remove all cards from the FEDAL tester. Insert the FEDAL out card to be tested in output slot Ø, with component side oriented per labels on the tester.

4. Check for shorts between supplies.

5. Use an ohm meter or buzzer to verify the following supply connections:

   +15 V to:
   Row B: Pin 41
   Row H: Pins 4, 18, 32, 46
   Row L: Pins 4, 18, 32, 46
   Row W: Pins 4, 18, 32, 46

   -15 V, on version 1.2 boards (-15 V decoupling caps plugged in on component side):

   Row B: Pin 42
   Row H: Pins 3V, 12, 19V, 26, 35V, 40, 51V, 54
   Row L: Pins 12, 26, 40, 54
   Row W: Pins 3V, 12, 19V, 26, 35V, 40, 51V, 54

   -15 V, on version 1.3 boards (-15 V decoupling caps wrapped on wire side):

   Row B: Pin 42
   Row H: Pins 12, 26, 40, 54
   Row J: Pins 9, 23, 37, 51
   Row L: Pins 12, 26, 40, 54
   Row W: Pins 12, 26, 40, 54

   15 V RTN, on version 1.2 boards:

   Row B: Pins 38, 39
   Row C: Pins 41, 42
   Row H: Pins 5, 70, 19, 23G, 33, 39G, 47. 55G
   Row L: Pins 5, 19, 33, 47
   Row P: Pins 7G, 23G, 39G, 55G
   Row W: Pins 5, 7G, 19, 23G, 33, 39G, 47, 55G

15 V RTN, on version 1.3 boards:

Row B: Pins 38, 39
Row C: Pins 41, 42
Row H: Pins 5, 19, 33, 47
Row J: Pins 8, 22, 36, 50
Row L: Pins 5, 19, 33, 47
Row W: Pins 5, 19, 33, 47

+5 V to:

Row B: Pins 2, 11, 20, 29, 40, 45
Row D: Pins 1, 12, 23, 45
Row F: Pins 1, 12, 23, 34, 45
Row P: Pins 1, 12, 23, 34, 45
Row S: Pins 1, 12, 23, 34, 45
Row U: Pins 1, 12, 23, 45

5 V RTN to:

Row C: Pins 9, 18, 27, 36, 38, 39, 54
Row E: Pins 10, 21, 32, 54
Row G: Pins 10, 21, 32, 43, 54
Row R: Pins 10, 21, 32, 43, 54
Row T: Pins 10, 21, 32, 43, 54
Row V: Pins 10, 21, 32, 54

6. Remove the card from the tester. Connect the supplies. Turn the supplies on and verify that less than 50 milliamps are drawn by the tester with no cards plugged in.

7. Insert card under test, with no components inserted, into slot Out-Ø. Verify that supply current does not increase.

8. Remove the card and insert all discrete components. Plug the card back in, and verify that the currents do not change. If they do change, a polarized cap may be in backwards, or a capacitor may be shorted.

9. Turn power off, and install the remainder of the components. Remember to cut off pin 23 on the DAC 80's.

10. Turn power on and verify the following currents (above tester requirements):

+15 volts ..... 200 mA to 360 mA
-15 volts ..... 160 mA to 240 mA
+5 volts ..... 400 mA to 900 mA

If any of these are out of range, find the faulty component, or see if any components are plugged in backwards. Current consumption by chip is as follows:

10. Continued:

LS138	6.3	typ. ...	10 max	(5 V only)
LS240	20	typ. ...	44 max	(5 V only)
LS377	20	typ. ...	35 max	(5 V only)
HS DAC80 ...	0.0	typ. ...	0 max	(+5 V)
HS DAC80 ...	30	typ. ...	36 max	(+15 V)
HS DAC80 ...	18	typ. ...	24 max	(-15 V)

11. Turn power off. Insert "Test 3" ROM into the microprocessor board and plug this board into its socket. Turn power back on.

12. Connect scope channel 1 to FEDAL OUT board, pin C5. Trigger scope on channel 1. Verify the presence of 0.75 microsecond, low-going pulses on this pin. If not, see that it is present on FEDAL OUT pin A-39. If it is, FEDAL OUT wiring is at fault. If not, something is wrong with the microprocessor board.

13. Verify the presence of 0.5 microsecond pulses on:

Row B: Pins 4, 5, 6, 12, 13, 14, 15, 21,
22, 23, 24, 25, 26, 27, 30,
31, 32, 33, 34, 35, 36

Row C: Pins 26, 35

If not, check wiring and address bits A0 to A4 per sheet 1 of the schematic.

14. Verify that the LED's on the tester's test board, locations B7, 12, respond to the toggle switches at locations B8, 13. If not, check wiring per bottom half of sheet 1 of schematic.

15. Ground the scope probe on 15 V TRN, at pin B38. Verify the presence of smooth, negative going ramps that go from +10 V to -10 V in about 780 milliseconds, from each of the DAC's. These ramps can be found on the test board location B01. Pins 1 thru 12 correspond to ANAOUT-0 thru 11 on the schematic sheets 2 thru 7. If the ramps are not smooth, bits are getting lost between the input buffer at 1E (sheet 1) and the DAC's (sheets 2 thru 7), or the DAC's are faulty. With the scope set at a sweep rate of 0.1 sec per div., 5 V per div., internal rising edge triggered, and input DC coupled, the entire ramp can be viewed. Increasing the sweep rate to 200 microseconds per division, 10 millivolts per division, same trigger, and viewing the test points through a 1000 pF capacitor, allows viewing of the small steps in the ramp. Since they are viewed through a 1000 pF capacitor, they appear as small, positive going ramps. The step size should be about 4.9 millivolts in amplitude and 200 microseconds in duration. The "thick grass" hash around the ramps should not be more than 10 to 15 millivolts peak. If it is, digital wires may be coupling into analog wires and should be rearranged, or additional supply decoupling may be required. Large spikes at the transitions are expected; the grass we are concerned about is between the transitions only.

TEST PROCEDURE FOR FRONT-END DATA LINK CHASSIS

R. Lacasse

August 22, 1983

Introduction

This test procedure is to be used to debug Front-End Data Link Chassis using a set of cards that are known to be good. Thus, most problems should be attributable to wiring errors. A wiring list is included in Appendix A for reference.

1. Set-Up

a. Locate the chassis to be tested on two 1" x 1" x 1' pieces of wood to allow air to circulate under the bottom cover.

b. Connect the chassis to the Front-End Data Link Tester by means of the four multiconductor cables.

c. Connect a 5 volt supply to the tester and power on. Verify that it supplies $\leq$ 500 mA.

d. Connect the resistors at B26 and B27 on the tester to the 15 V Ret. binding post.

e. Connect AC power to the chassis. Lower the chassis front panel. Power on and measure the supply voltages on the terminal board. Terminal assignments are shown on page A-9. Supplies should be within 10 millivolts of nominal. If not, adjust the supplies

2. Supply Bus Tests

Power off. Remove the part of the rear cover that protects the connector pins. Power on. Verify the presence of +5 V on pins 2 and 100, 0 V on pins 1, 99, and 69, +15 V on pin 61, and -15 V on pin 65 of each slot.

3. Supply Current Tests

Power off. Insert a tested set of cards into the card cage. Power on and verify the following supply currents on the terminal board, using a Simpson multimeter or equivalent:

Supply	Typ (A)	Max (A)
+5 V	2.7	4.0
+15 V	0.8	1.2
-15 V	0.5	0.8

4. Link Continuity Tests

   a. Verify that, with the chassis' front panel select switch in STATUS position, both local and remote displays read 00.

   b. Power off. Remove the component carrier at A15 on the tester. Power on. The Status Display should read

      Local 01
      Remote ... Blank

      for about one second. Then it should read all zeros.

   c. With power on, pull the component carrier at A6. The local display should read 01 in about 1 second.

   d. Replace A6 and A15. The display should go to all zeros immediately.

5. Digital I/O Test

   Set Digi-Switch 4 on the TESTER to 70. Verify that the LED's in location B7, B12, B17 and B22 respond to the switches in locations B8, B13, B18, and B23. If not, check the wiring to the Digital I/O connector.

6. Analog I/O Tests

   Leave all analog inputs (B26 and B27) tied to 15 V RTN through the 10 KΩ resistors. Apply a 20 V p-p, 1 Hz, sinusoid directly (i.e., on the other side of the resistor) to each of the analog inputs in turn. Verify that the sampled version of the signal appears on the appropriate analog output and that the analog output goes to 0 V when the signal is applied to the next input channel. Analog I/O is summarized in Table 1.

7. User Limit Tests

   a. Select channel 00 on the chassis front panel monitor with the digi-switch. Connect a variable DC supply, referenced to 15 V RTN, to Analog Input 0 (B2601). The front panel meter should read the supply voltage. When the supply voltage is turned up to $\geq$ +9.98 V, $\pm$ .02 V, the front panel meter should blink.

   b. Set switch 4 on the Tester to 80. Verify that the LED at B1708-B1709 is off when the front panel meter blinks, and on when it does not (i.e., as the DC input supply is adjusted over and under the limit). Verify the same for analog input channels 1-15. The mapping of channels to LED's is shown in Table 3.

8. Chassis Temperature

   After at least a 30 minute warmup period, measure the air temperature inside the chassis in between the cards and in the vicinity of the supplies. The temperature should be $\leq$ 45°C.

9. Burn the unit in for 4 to 7 days and repeat steps 1 through 8.

TABLE 1: Analog I/O Locations

Channel	Analog Input	Analog Output
0	B26-01	B01-01
1	-02	-02
2	-03	-03
3	-04	-04
4	-05	-05
5	-06	-06
6	-07	-07
7	-08	-08
8	-09	-09
9	-10	-10
10	-11	-11
11	-12	-12
12	-13	B02-01
13	-14	-02
14	-15	-03
15	-16	-04
16	B27-01	-05
17	-02	-06
18	-03	-07
19	-04	-08
20	-05	-09
21	-06	-10
22	-07	-11
23	-08	-12
24	-09	B03-01
25	-10	-02
26	-11	-03
27	-12	-04
28	-13	-05
29	-14	-06
30	-15	-07
31	-16	-08

TABLE 2: Tester Limit LED Locations vs. Analog Input

Analog Input	Limit LED Location
0	B17-08
1	-07
2	-06
3	-05
4	-04
5	-03
6	-02
7	-01
8	B22-08
9	-07
10	-06
11	-05
12	-04
13	-03
14	-02
15	-01

APPENDIX A

FRONT END DATA LINK CHASSIS WIRING

```
MEMBER NAME  FECALWI
**********************************************************************
*                                                                    *
*                          FRONT END DATA LINK                       *
*                            CHASSIS WIRING                          *
*                                                  REV.: 12-22-82    *
**********************************************************************

GENERAL INFORMATION:
1-CONNECTOR DESIGNATIONS:
  S1 THROUGH S5: CARD CAGE WIRE WRAP 100 PIN CONNECTIORS, WITH S1 AT TOP
  TB: TERMINAL BOARD ON FLOOR OF CHSSSIS
  P1 THROUGH P5: REAR PANEL ELCC AND D CONNECTORS WITH P1 AT TOP
  P6: FRONT PANEL TOP DIGISWITCH,MOST SIGNIFICANT DIGIT
  P7: FRONT PANEL TOP DIGISWITCH,LEAST SIGNIFICANT DIGIT
  P8: FRONT PANEL BOTTOM DIGISWITCH, MOST SIGNIFICANT DIGIT
  P9: FRONT PANEL BOTTOM DIGISWITCH, LEAST SIGNIFICANT DIGIT
  P10:FRONT PANEL DIGITAL PANEL METER CONNECTOR
  P11: FRONT PANEL ANALOG PANEL METER CONNECTORS
  D1: FRONT PANEL DISPLAY, LEFT-MOST DIGIT
  D2: FRONT PANEL DISPLAY, SECOND DIGIT FROM LEFT
  D3: FRONT PANEL DISPLAY, THIRD DIGIT FROM LEFT
  D4: FRONT PANEL DISPLAY, RIGHT-MOST DIGIT
  SW: FRONT PANEL DISPLAY SELECTOR SWITCH, PER SEPARATE SCHEMATIC.
2-INSTALL DIODES ON FRONT PANEL DIGISWITCHES
3-INSTALL ALL BUS-BARS ON CARD CAGE, AND CHECK RESISTANCE BETWEEN BARS.
4-DO THE REST OF THE WIRING IN ANY CONVENIENT ORDER.
5-BUZZ OUT ALL WIRING.
6-KEEP ANALOG WIRING AWAY FROM DIGITAL WIRING. SIGNALS NAMES WHOSE
  FIRST LETTER IS 'A' ARE ANALOG SIGNALS.
```

```
MEMBER NAME  FEDALWI
**********************************************************************
*                                                                    *
*  CARD CAGE WIRING LIST                                             *
*                                                                    *
**********************************************************************

GENERAL INFORMATION
 1-SLOTS ARE NUMBERED 1 THROUGH 5 FROMTOP TO BOTTOM.
 2-PINS ARE CALLED OUT AS FOLLOWS: SX-YZ, WHERE X IS THE
   SLOT NUMBER, 1 THROUGH 5, AND YZ IS THE PIN NUMBER.
   FOR EXAMPLE S4-38 REFERS TO SLOT 4 PIN 38.
3-A "B" IN THE TO/FROM COLUMN MEANS THAT THE SIGNAL ON
  THAT PIN IS BUSSED VERTICALLY THROUGH THE CARD CAGE.
  FOR EXAMPLE ,A "B" IN COLUMN 5 MEANS THAT ALL PINS
  5 AND 6 IN THE CARD CAGE ARE BUSSED TOGETHER WITH A
  BUS BAR. PIN 6 IS INCLUDED BECAUSE IT IS OPPOSITE
  PIN 5 AND THERE IS NO WAY TO BUS PINS 5 TOGETHER WITHOUT
  ALSO INCLUDING PINS 6.
 4-A "N1" IN THE TO/FROM COLUMN MEANS THAT THE SIGNAL ON THAT PIN
   IS BUSSED VERTICALLY THROUGH THE CARD CAGE, FROM THE ODD
   PINS OF SLOT 2 DOWNWARD, I.E., SLOT 1 AND THE EVEN PINS OF
   SLOT 1 ARE NOT INCLUDED IN THIS BUS.
 5-A LOWER CASE LETTER IS INDICATED BY A "LESS THAN" SIGN, "<".
   FOR EXAMPLE LOWER CASE L IS PRINTED <L.
```

MEMBER NAME FEDALWI

```
*****************************************************************************
*     SLOT 1 WIRING                SLCT NAME: ANIN                          *
*****************************************************************************
```

NAME	PIN	TO/FROM		NAME	PIN	TO/FROM
5V RET	1	RUNION GND BUS		+5V	2	RUNION VCC BUS
5V RET	3	RUNION GND BUS		+5V	4	RUNION VCC BUS
D7	5	B		D7-	6	B
D6	7	B		D6-	8	B
D5	9	B		D5-	10	B
D4	11	B		D4-	12	B
D3	13	B		D3-	14	B
D2	15	B		D2-	16	B
D1	17	B		D1-	18	B
D0	19	B		D0-	20	B
	21				22	
	23				24	
	25			ADCSEL-	26	S1-26
5V RET	27	RUNION GND BUS		5V RET	28	RUNION GND BUS
A1-	29	B		A1-	30	B
A0-	31	B		A0-	32	B
	33				34	
G-	35	B		G-	36	B
	37				38	
ANIN-0	39	P1-<L		ANIN-1	40	P1-<K
ANIN-2	41	P1-<J		ANIN-3	42	P1-<H
ANIN-4	43	P1-<F		ANIN-5	44	P1-<E
ANIN-6	45	P1-<D		ANIN-7	46	P1-<C
	47				48	
5V RET	49	RUNION GND BUS		5V RET	50	RUNION GND BUS
	51				52	
ANIN-8	53	P1-<B		ANIN-9	54	P1-<A
ANIN-10	55	P1-Z		ANIN-11	56	P1-Y
ANIN-12	57	P1-X		ANIN-13	58	P1-W
ANIN-14	59	P1-V		ANIN-15	60	P1-U
+15V	61	B		+15V	62	B
	63				64	
-15V	65	B		-15V	66	B
PSELDOGND	67	J1-<M			68	
15V RET	69	B		15V RET	70	B
	71				72	
5V RET	73	RUNION GND BUS		5V RET	74	RUNION GND BUS
	75				76	
ANIN-16	77	P1-T		ANIN-17	78	P1-S
ANIN-18	79	P1-R		ANIN-19	80	P1-P
ANIN-20	81	P1-N		ANIN-21	82	P1-M
ANIN-22	83	P1-L		ANIN-23	84	P1-K
ANIN-24	85	P1-J		ANIN-25	86	P1-H
ANIN-26	87	P1-F		ANIN-27	88	P1-E
ANIN-28	89	P1-D		ANIN-29	90	P1-C
ANIN-30	91	P1-B		ANIN-31	92	P1-A
	93				94	
	95				96	
5V RET	97	RUNION GND BUS		+5V	98	RUNION VCC BUS
5V RET	99	RUNION GND BUS		+5V	100	RUNION VCC BUS

MEMBER NAME 'FECALWI

```
*****************************************************************************
*      SLOT 2 WIRING                          SLOT NAME:CONTROLLER            *
*****************************************************************************
```

NAME	PIN	TO/FROM	NAME	PIN	TO FROM+----6----
5V RET	1	RUNION GND BUS	+5V	2	RUNION VCC BUS
5V RET	3	RUNION GND BUS	+5V	4	RUNION VCC BUS
D7	5	B	D7	6	B
D6	7	B	D6	8	B
D5	9	B	D5	10	B
D4	11	B	D4	12	B
D3	13	B	D3	14	B
D2	15	B	D2	16	B
D1	17	B	D1	18	B
D0	19	B	D0	20	B
A4-	21	N1		22	
A3-		N1		24	
A2-	25	N1	ADCSEL-	26	S1-26
C RET	27	RUNION GND BUS	5V RET	28	RUNION GND BUS
A1-	29	B	A1-	30	B
A0-	31	B	A0-	32	B
	33			34	
G-	35	B	G-	36	B
OUTSEL0-	37	S3-39		38	
OUTSEL1-	39	S4-39		40	
OUTSEL2-	41	S5-39		42	
LLATEN-	43	D1-5		44	
RLATEN-	45	D3-5		46	
RXC+	47	P3-A	RXC-	48	P3-B
5V RET	49	RUNION GND BUS	5V RET	50	RUNION GND BUS
RX1+	51	P3-E	RX1-	52	P3-F
	53			54	
TXC+	55	P3-C	TXC-	56	P3-D
TX1+	57	P3-J	TX1-	58	P3-K
	59			60	
+15V	61	B	+15V	62	B
	63			64	
-15V	65	B	-15V	66	B
	67			68	
15V RET	69	B	15V RET	70	B
DSR	71	P2-6	SELSWEN-	72	FRONT PAN. SEL. SW.
5V RET	73	GND BUS;P2-1;P2-7	5V RET	74	RUNION GND BUS
RS232IN+	75	P2-3	RS232IN-	76	S2-74
RS232OUT+	77	P2-2	CSW0SEL	78	P6-C;P7-C
RS232OUT-	79	NO CONNECTION	CS1SEL9	80	P8-C;P9-C
MSD-8	81	P3-H;P6-8;P8-8	CSW2SEL-	82	P3-U
MSD-4	83	P3-L;P6-4;P8-4	CSW3SEL-	84	P3-V
MSD-2	85	P3-M;P6-2;P8-2	CSW4SEL-	86	P3-W
MSD-1	87	P3-N;P6-1;P8-1	CSW5SEL-	88	P3-X
LSD-8	89	P3-P;P7-8;P9-8	TSW3SEL-	90	P3-Y
LSD-4	91	P3-R;P7-4;P9-4	TSW2SEL-	92	P3-Z
LSD-2	93	P3-S;P7-2;P9-2;SW	TSW1SEL-	94	P3-<A
LSD-1	95	P3-T;P7-1;P9-1;SW	TSW0SEL-	96	P3-<B
5V RET	97	RUNION GND BUS	+5V	98	RUNION VCC BUS
5V RET	99	RUNION GND BUS	+5V	100	RUNION VCC BUS

MEMBER NAME FEDALWI

```
*****************************************************************************
*     SLOT 3 WIRING                          SLCT NAME: OUTC                *
*****************************************************************************
```

NAME	PIN	TO/FROM	NAME	PIN	TO FROM
5V RET	1	RUNION GND BUS	+5V	2	RUNION VCC BUS
5V RET	3	RUNION GND BUS	+5V	4	RUNION VCC BUS
D7	5	B	D7	6	B
D6	7	B	D6	8	B
D5	9	B	D5	10	B
D4	11	B	D4	12	B
D3	13	B	D3	14	B
D2	15	B	D2	16	B
D1	17	B	D1	18	B
D0	19	B	D0	20	B
A4-	21	N1	A4-	22	N1
A3-		N1	A3-	24	N1
A2-	25	N1	A2-	26	N1
5V RET	27	RUNION GND BUS	5V RET	28	RUNION GND BUS
A1-	29	B	A1-	30	B
A0-	31	B	A0-	32	B
	33			34	
Q-	35	B	Q-	36	B
	37			38	
OUTSEL0-	39	S2-37		40	
DOUT15-0	41	P3-<W	DOUT14-0	42	P3-<X
DOUT13-0	43	P3-<Y	DOUT12-0	44	P3-<Z
DOUT11-0	45	P3-AA	DOUT-10	46	P3-BB
DOUT9-0	47	P3-CC	DOUT8-0	48	P3-DD
5V RET	49	RUNION GND BUS	5V RET	50	RUNION GND BUS
DOUT7-0	51	P3-EE	DOUT6-0	52	P3-FF
DOUT5-0	53	P3-HH	DOUT4-0	54	P3-JJ
DOUT3-0	55	P3-KK	DOUT2-0	56	P3-LL
DOUT1-0	57	P3-MM	DOUT0-0	58	P3-NN
	59			60	
+15V	61	B	+15V	62	B
	63			64	
-15V	65	B	-15V	66	B
	67			68	
15V RET	69	B	15V RET	70	B
ANOUT1-0	71	P4-A	15VRET	72	P4-B TWIST 71-72
5V RET	73	RUNION GND BUS	5V RET	74	RUNION GND BUS
ANOUT 1-0	75	P4-C	15VRETN-	76	P4-D TWIST 75-76
ANOUT2-0	77	P4-E	15VRETL	78	P4-F TWIST 77-78
ANOUT3-0	79	P4-J	15VRET	80	P4-K TWIST 79-80
ANOUT4-0	81	P4-L	15VRET	82	P4-M TWIST 81-82
ANOUT5-0	83	P4-N	15VRET	84	P4-P TWIST 83-84
ANOUT6-0	85	P4-R	15VRET	86	P4-S TWIST 85-86
ANOUT7-0	87	P4-U	15VRET	88	P4-V TWIST 87-88
ANOUT8-0	89	P4-W	15VRET	90	P4-X TWIST 89-90
ANOUT9-0	91	P4-Y	15VET	92	P4-Z TWIST 91-92
ANOUT10-0	93	P4-<A	15VRET	94	P4-<B TWIST 93-94
ANOUT11-0	95	P4-<C	15VRET	96	P4-<D TWIST 95-96
5V RET	97	RUNION GND BUS	+5V	98	RUNION VCC BUS
5V RET	99	RUNION GND BUS	+5V	100	RUNION VCC BUS

MEMBER NAME FECALWI

****
* SLOT 4 WIRING SLOT NAME: OUT1 *
****

NAME	PIN	TO/FROM	NAME	PIN	TO FROM
5V RET	1	RUNION GND BUS	+5V	2	RUNION VCC BUS
5V RET	3	RUNION GND BUS	+5V	4	RUNION VCC BUS
D7	5	B	D7	6	B
D6	7	B	D6	8	B
D5	9	B	D5	10	B
D4	11	B	D4	12	B
D3	13	B	D3	14	B
D2	15	B	D2	16	B
D1	17	B	D1	18	B
D0	19	B	D0	20	B
A4-	21	N1	A4-	22	N1
A3-	23	N1	A3-	24	N1
A2-	25	N1	A2-	26	N1
5V RET	27	RUNION GND BUS	5V RET	28	RUNION GND BUS
A1-	29	B	A1-	30	B
A0-	31	B	A0-	32	B
	33			34	
Q-	35	B	Q-	36	B
	37			38	
OUTSEL1-	39	S2-39		40	
DOUT15-1	41	P3-<C	DOUT14-1	42	P3-<D
DOUT13-1	43	P3-<E	DOUT12-1	44	P3-<F
DOUT11-1	45	P3-<H	DOUT10-1	46	P3-<J
DOUT9-1	47	P3-<K	DOUT8-1	48	P3-<L
5V RET	49	RUNION GND BUS	5V RET	50	RUNION GND BUS
DOUT7-1	51	P3-<M	DOUT6-1	52	P3-<N
DOUT5-1	53	P3-<P	DOUT4-1	54	P3-<R
DOUT3-1	55	P3-<S	DOUT2-1	56	P3-<T
DOUT1-1	57	P3-<U	DOUT0-1	58	P3-<V
	59			60	
+15V	61	B	+15V	62	B
	63			64	
-15V	65	B	-15V	66	B
	67			68	
15V RET	69	B	15V RET	70	B
ANOUT0-1	71	P4-<K	15VRET	72	P4-<L TWIST 71-72
5V RET	73	RUNION GND BUS	5V RET	74	RUNION GND BUS
ANOUT 1-1	75	P4-<M	15VRET	76	P4-<N TWIST 75-76
ANOUT2-1	77	P4-<P	15VRET	78	P4-<R TWIST 77-78
ANOUT3-1	79	P4-<S	15VRET	80	P4-<T TWIST 79-80
ANOUT4-1	81	P5-A	15VRET	82	P5-B TWIST 81-82
ANOUT5-1	83	P5-C	15VRET	84	P5-D TWIST 83-84
ANOUT6-1	85	P5-E	15VRET	86	P5-F TWIST 85-86
ANOUT7-1	87	P5-J	15VRET	88	P5-K TWIST 87-88
ANOUT8-1	89	P5-L	15VRET	90	P5-M TWIST 89-90
ANOUT9-1	91	P5-N	15VRET	92	P5-P TWIST 91-92
ANOUT10-1	93	P5-R	15VRET	94	P5-S TWIST 93-94
ANOUT11-1	95	P5-U	15VRET	96	P5-V TWIST 95-96
5V RET	97	RUNION GND BUS	+5V	98	RUNION VCC BUS
5V RET	99	RUNION GND BUS	+5V	100	RUNION VCC BUS

MEMBER NAME FEDALWI

****
* SLOT 5 WIRING SLOT NAME: OUT2 *
****

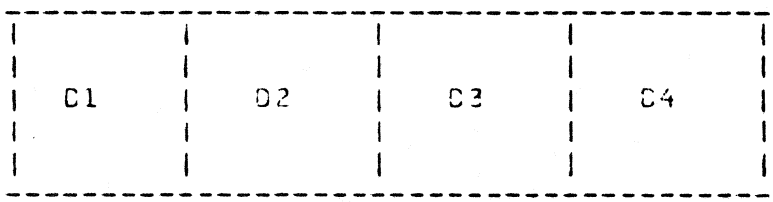
NAME	PIN	TO/FROM	NAME	PIN	TO FROM
5V RET	1	RUNION GND BUS	+5V	2	RUNION VCC BUS
5V RET	3	RUNION GND BUS	+5V	4	RUNION VCC BUS
D7	5	B; D1-3	D7	6	B
D6	7	B; D1-2	D6	8	B
D5	9	B; D1-1	D5	10	B
D4	11	B; D1-8	D4	12	B
D3	13	B; D2-3	D3	14	B
D2	15	B; D2-2	D2	16	B
D1	17	B; D2-1	D1	18	B
D0	19	B; D2-8	D0	20	B
A4-	21	N1	A4-	22	N1
A3-	23	N1	A3-	24	N1
A2-	25	N1	A2-	26	N1
5V RET	27	RUNION GND BUS	5V RET	28	RUNION GND BUS
A1-	29	B	A1-	30	B
A0-	31	B	A0-	32	B
	33			34	
Q-	35	B	Q-	36	P
	37			38	
OUTSEL2-	39	S2-41		40	
DOUT15-2	41		DOUT14-2	42	
DOUT13-2	43		DOUT12-2	44	
DOUT11-2	45		DOUT10-2	46	
DOUT9-2	47		DOUT8-2	48	
5V RET	49	RUNION GND BUS	5V RET	50	RUNION GND BUS
DOUT7-2	51		DOUT6-2	52	
DOUT5-2	53		DOUT4-2	54	
DOUT3-2	55		DOUT2-2	56	
DOUT1-2	57	D3-4	DOUT0-2	58	P10-A6
	59			60	
+15V	61	B	+15V	62	B
	63			64	
-15V	65	B	-15V	66	B
	67			68	
15V RET	69	B	15V RET	70	B
ANOUT0-2	71	P5-W	15VRET	72	P5-X TWIST 71-72
5V RET	73	RUNION GND BUS	5V RET	74	RUNION GND BUS
ANOUT 1-2	75	P5-Y	15VRET	76	P5-Z TWIST 75-76
ANOUT2-2	77	P5-<A	15VRET	78	P5-<B TWIST 77-78
ANOUT3-2	79	P5-<C	15VRET	80	P5-<D TWIST 79-80
ANOUT4-2	81	P5-<K	15VRET	82	P5-<L TWIST 81-82
ANOUT5-2	83	P5-<M	15VRET	84	P5-<N TWIST 83-84
ANOUT6-2	85	P5-<P	15VRET	86	P5-<R TWIST 85-86
ANOUT7-2	87	P5-<S	15VRET	88	P5-<T TWIST 87-88
ANOUT8-2	89	P10-A14	15VRET	90	P10-A15 TWIST 89-90
ANOUT9-2	91	P11+	15VRET	92	P11- TWIST 91-92
ANOUT10-2	93	P4-<Z	15VRET	94	P4-AA TWIST 93-94
ANOUT11-2	95	P4-BB	15VRET	96	P4-CC TWIST 95-96
5V RET	97	RUNION GND BUS	+5V	98	RUNION VCC BUS
5V RET	99	RUNION GND BUS	+5V	100	RUNION VCC BUS

MEMBER NAME FEDALWI

```
*****************************************************************************
*                                                                           *
*  LOCAL/REMOTE DISPLAY WIRING LIST                                         *
*                                                                           *
*****************************************************************************
```

DEFINITION OF SYMBOLS:

DIGIT NUMBERING AS VIEWED FROM OUTSIDE THE CHASSIS:



EXAMPLE: D3-4 REFERS TO PIN 4 ON THE THIRD DIGIT FROM THE LEFT

SIGNAL NAME	PIN	TO/FROM	COMMENTS
D5	D1-1	S5-9 ; D3-1	
D6	D1-2	S5-7 ; D3-2	
D7	D1-3	S5-5 ; D3-3	
BLANK1+	D1-4	D1-6	
BLANK2+	D2-4	D2-6	
BLANK3+	D3-4	D4-4 ; S5-57	
LLATEN-	D1-5	S2-43 ; D2-5	
5V RET	D1-6	TB-2 ; D2-6 ; D3-6 ; D4-6	USE STRANDED 22 AWG
+5V	D1-7	TB-4 ; D2-7 ; D3-7 ; D4-7	USE STRANDED 22 AWG
D4	D1-8	S5-11 ; D3-8	
D1	D2-1	S5-17 ; D4-1	
D2	D2-2	S5-15 ; D4-2	
D3	D2-3	S5-13 ; D4-3	
D0	D2-8	S5-19 ; D4-8	
RLATEN-	D3-5	S2-45 ; D4-5	

Terminal Board Wiring

Signal Name	TB-	To/From
Chassis Ground	1	TB-2
5 V Ret	2	TB-1; TB-3 5 V Supply Return DPM-A1 Card Cage 5 V Return Bus LED Displays, Pin 6
15 V Ret	3	TB-2 15 V Supply Return Card Cage 15 V Return Bus
+5 V	4	+5 V Supply DPM-A2 Card Cage +5 V Bus LED Displays, Pin 7
+15 V	5	+15 V Supply Card Cage +15 V Bus
-15 V	6	-15 V Supply Card Cage -15 V Bus
Spare	7	
AC Neutral	8	Front Panel Power Switch +5 V Supply ± 15 V Supply
AC Hot	9	Rear Panel AC Connector +5 V Supply ± 15 V Supply

P 10

Digital Panel Meter Connector Wiring

Signal Name	DPM-	From/To	
5 V Ret	A1	TB-2	
+5 V	A2	TB-4	
Mon. -Ø	A14	Mon Ø BNC; S5-89	Twisted
15 V Ret	A15	BNC Ret; S5-90	
Decimal Pt.	B7	B10	

--- Others Not Used ---

P 11

Analog Panel Meter Wiring

Signal Name	APM	From/To
Mon-1	+	S5-91; Mon 1 BNC
15 V Ret	-	S5-92; BNC Ret

Select Switch Wiring