NOTICE
PORTIONS OF THIS REPORT ARE ILLEGIBLE.
It has been reproduced from the best
available copy to permit the broadest
possible availability.

A New 1-k PROM for the Coincidence- Counter Electronics Package

J. E. Swansen

DISCLAIMER

LA--9190-MS

DE82 009541

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

ABSTRACT	1
I. INTRODUCTION	1
II. BRIEF DESCRIPTION OF THE NEW OPERATING MODES	2
A. Front-Panel Control Mode	2
B. Computer Control Mode	2
C. Terminal Control Mode	3
III. OPERATING INSTRUCTIONS FOR COMPUTER CONTROL MODE	3
IV. OPERATING INSTRUCTIONS FOR TERMINAL CONTROL MODE	6
V. STATUS WORD DESCRIPTION	7
VI. ACKNOWLEDGMENTS	8
APPENDIX A: INSTALLATION AND ADDITIONAL DETAILS ON USE OF THE NEW PROM . .	9
APPENDIX B: MICROPROCESSOR PROGRAM FLOW CHART	13
APPENDIX C: ASSEMBLY LANGUAGE LISTING	23
APPENDIX D: FORTRAN-CALLABLE DRIVER AND FORTRAN TEST PROGRAM	37

A NEW 1-k PROM FOR THE COINCIDENCE COUNTER ELECTRONICS PACKAGE

by

J. E. Swansen

ABSTRACT

A new programmable read-only memory (PROM) for the Los Alamos-designed neutron coincidence electronics package is described. The new 1-k PROM allows remote control of the electronics by a computer or a remote terminal through an RS-232 serial data port. No modifications of the existing unit are required.

I. INTRODUCTION

The portable high-level neutron coincidence counter (HLNCC), the active well coincidence counter (AWCC), and a variety of in-plant neutron counters have been accepted as useful tools for the *nondestructive* assay of fissionable materials. The electronics package used to operate these detectors consists of a high-voltage power supply, six amplifiers, a shift-register coincidence circuit, and four scalers. This package was designed at Los Alamos National Laboratory and is commercially available through IRT Corporation, San Diego, California.

Although the electronics package was originally designed as a stand-alone instrument with only a front-panel display, modifications were added after several users showed interest in controlling the instrument remotely. The instrument was first modified to send data to a Hewlett Packard HP-97 programmable calculator for local data processing. This modification was implemented by interfacing a small Motorola MC6800 microprocessor system to the existing hardware. In addition to providing data readout capability, this interface allows processor control of the basic hardware functions: start, stop, and reset. A recycle mode was added, and an RS-232 serial data port was also

provided for auxiliary data output. Both of these features are under the control of the MC6800 software. When originally implemented, the RS-232 port was set up only to output data, but hardware was provided so that this port could later be used to input data. The new 1-k programmable read-only memory (PROM) described in this report takes advantage of this provision. Remote control is accomplished by inputting ASCII command characters and outputting data through the RS-232 port. Remote control is possible from any RS-232-compatible device, such as a stand-alone data terminal or a computer that has been configured to accept a second terminal.

Section II of this report provides a brief description of the front-panel, computer, and terminal control modes. The computer and terminal control modes are described more fully in Secs. III and IV. Installation of the new PROM and additional details on its use are given in Appendix A. Appendix B shows a flow chart of the microprocessor program, and Appendix C provides an assembly language listing. Appendix D lists a FORTRAN-callable driver and a FORTRAN test program to illustrate operation of the new PROM under computer control.

II. BRIEF DESCRIPTION OF THE NEW OPERATING MODES

A. Front-Panel Control Mode

This mode is the default mode when the power is turned on. The front-panel control mode operates in a similar manner as it did with the previous PROM, with minor exceptions as noted in Appendix A. The front panel controls the operation of the counter. Readout is directed to both the HP-97 calculator and the RS-232 serial port. The instrument must be in the front-panel control mode to enter either the computer control mode or the terminal control mode. While becoming familiar with the new PROM, the user can always return to front-panel control by pressing the restart push button on the rear panel or by turning the power off momentarily.

B. Computer Control Mode

This mode allows remote control of the electronics package through the RS-232 serial port. The serial port setup is described in Appendix A. The following functions may be controlled remotely.

- S - Start
- H - Stop (halt)
- Z - Clear scalers and timer
- R - Readout data to computer
- P - Readout to HP-97 calculator
- X - Request status
- I - Set status interrupt flag (I flag)
- F - Return to front-panel control.

The control functions are achieved by transmitting the appropriate command from the computer to the electronics package. Computer control is designed for ease of control from the remote computer. The electronics package may be operated in a passive mode where information is returned to the computer only when requested or in a pseudo-interrupt mode where a flag (I flag) signals the computer when there is a change in status. Commands from the computer are not echoed back to the computer. All data sent to the computer are terminated by a carriage return/line feed (CRLF).

C. Terminal Control Mode

This mode allows a terminal to function as a remote front panel for the electronics package. The terminal control mode operates similarly to the computer control mode but provides more information to the operator. Commands to the electronics package are the same as listed above for the computer control mode, but are all ASCII control characters (the CTRL key and letter key are depressed simultaneously). Commands echoed back to the terminal are preceded by T and followed by CRLF. Invalid control characters are echoed in the same manner, but are followed by a question mark and bell. All ASCII printing characters are echoed as is, allowing the operator to type identification or other text on the terminal without inadvertently interfering with operation of the electronics package.

III. OPERATING INSTRUCTIONS FOR COMPUTER CONTROL MODE

To enter the computer control mode, transmit ^ from the computer while in the front-panel mode. The electronics package will respond by returning to the computer: CRLF version number CRLF. This is followed by an 8-character status

word CRLF (see Sec. V). The electronics package then waits for one of the following commands from the computer.

- S - Start
- H - Stop (halt)
- Z - Clear scalers and timer
- R - Readout data to computer
- P - Readout data to HP-97 calculator
- X - Request status
- I - Set status interrupt flag (I flag)
- F - Return to front-panel control

All other inputs are ignored. Also, the electronics package will not echo command characters back to the computer while in the computer control mode. Upon receiving one of the above commands, the electronics package will execute it with the following conditions.

- (1) The I command allows the computer to be advised of changes in status. When the I flag is set, the next status change causes the electronics package to send a single character with CPLF to the computer. This character has a numerical value between 0 and 8, representing the status bit that has changed. A numerical value of 0 indicates no status change; it is transmitted only after an HP-97 calculator readout. The other status values are defined in Sec. V. Normally, the I command would be used only before a timeout or a data readout to the HP-97 calculator. The I flag is cleared after a status change. The I command must be sent again before the next desired status change.
- (2) If the electronics package had previously timed out, it must be reset (Z) before it can be restarted (S).
- (3) Readout commands R and P are operable only when the scalers are stopped; otherwise, they are ignored.
- (4) The F command allows the computer to return control to the front panel.

Four types of data are sent to the computer:

- (1) Version number: CRLF V 2.1 12 Dec 80 CRLF (19 characters). This occurs only once upon entering the computer control mode.
- (2) I flag response: a single ASCII digit 0 through 8 CRLF (3 characters). This occurs only if the I flag has been set and there is a status change.

(3) Status word: xxxxxxxx CRLF (10 characters). (See Sec. V for detailed description.) This status word is sent to the computer under the following conditions.

(a) Immediately after version number upon entering computer control mode.

(b) When requested by the X command.

(4) Data String: time, totals, R+A, A CRLF (57 characters). Data are all numeric or space characters. This data string is sent to the computer if an R readout command has been received by the electronics package when it is stopped. Note that (R+A)-A is not sent in the computer control mode.

The format for these four types of data is given in Table I.

TABLE I
DATA FORMAT FOR COMPUTER CONTROL MODE^a

<u>Data Type</u>	<u>Total Character Count</u>
Version Number: CRLF V-2.1-12-DEC-80 CRLF	19
I Flag: x CRLF	3
Status Word: xxxxxxx CRLF	10
Data Readout: ^b <div style="display: flex; align-items: center; justify-content: space-around;"> xxx-xxx ---x-xxx-xxx-xxx ---x-xxx-xxx-xxx ---x-xxx-xxx-xxx </div> <div style="display: flex; align-items: center; justify-content: space-around; margin-top: 5px;"> Time Totals R+A A </div>	57

^ax denotes a numeric character; - denotes a space character.

^bLeading zeroes are replaced by spaces.

IV. OPERATING INSTRUCTIONS FOR TERMINAL CONTROL MODE

The terminal control mode is used for remotely controlling the electronics package from a terminal only. It differs from the computer control mode in that the control characters are used as commands from the terminal to the electronics.

To enter the terminal control mode, type CTRL T on the keyboard while in the front-panel mode. The electronics package responds by returning CRLF V 2.1 12 Dec 80 CRLF and an 8-character status word CRLF. The electronics package then waits for one of the following input commands from the terminal.

CTRL S - Start

CTPL H - Stop (halt)

CTPL Z - Clear scalars, timer, and fault condition

CTRL P - Readout data to terminal

CTRL P - Readout data to HP-97 calculator

CTRL X - Request status

CTRL F - Return to front-panel control

CTPL I - Set status interrupt flag (I flag)

Valid commands are echoed preceded by T and followed by CRLF. Invalid control characters are echoed in the same manner, but are followed by a question mark and bell. Printing characters do not affect operation of the electronics package, allowing typing of headings or other text on the terminal.

Upon receiving one of the above commands, the electronics package will execute it with the following conditions.

- (1) The operator may be advised of changes in status, if desired, by setting the status interrupt flag (CTRL I) before any anticipated status change, such as completion of count time or calculator readout. (These two operations require time to complete; if the terminal is located some distance away, the operator will know when to continue.) When the I flag is set, the next status change causes the status character to be printed on the terminal and the I flag to be cleared. See Sec. III, item (1), for additional details on the I command.
- (2) If the scalars had previously timed out, the electronics package must be reset (CTRL Z) before it can be started (CTRL S). If stopped for any other reason it may be restarted with a CTRL S.

- (3) Readout commands CTRL R and CTRL P are operable only when the electronics package is stopped.
- (4) The command CTRL F causes the terminal to return control to the front panel of the electronics package.
- (5) The command CTRL R causes data to be printed on the terminal in the following format.

Time, totals, R+A, A, (R+A)-A

Leading zeros are suppressed and digits are separated into groups of three by spaces for ease in reading.

V. STATUS WORD DESCRIPTION

The status word may be requested by the X command at any time except during readout. It is an 8-character word, each character being either an ASCII 1 or 0. Each character's position (1-8, reading from left to right) within the word determines its function, as shown in Table II. The status word is terminated by CRLF.

TABLE II
STATUS WORD DEFINITIONS

<u>Character Position^a and Function</u>	<u>Status</u>
1 - Run/stop	1 = stopped, 0 = counting
2 - Time out	1 = count terminated by time out
3 - Fault	1 = fault condition set
4 - Stop push button	1 = button is depressed
5 - Reset push button	1 = button is depressed
6 - Start push button	1 = button is depressed
7 - Manual/recycle switch	1 = manual, 0 = recycle
8 - Readout push button	1 = button is depressed

^aCharacter position within word (12345678 CRLF).

VI. ACKNOWLEDGMENTS

Parts of the microprocessor program described in this report were developed by M. S. Krick of Los Alamos for the stand-alone version of the electronics package. S. Johnson provided the FORTRAN-callable driver and the FORTRAN test program. Discussions with N. Ensslin of Los Alamos were also helpful.

APPENDIX A

INSTALLATION AND ADDITIONAL DETAILS ON USE OF THE NEW PROM

I. INSTALLATION

This operation requires no modification of the existing circuit board.

- (1) Remove ac power cord from instrument.
- (2) Remove top cover.
- (3) Remove screw from clamp at rear of printed circuit board assembly on right side, then swing clamp back out of the way.
- (4) Remove plastic pin from hinge assembly.
- (5) Pivot board assembly to upright position and reinstall plastic pin to hold boards in this position. Save packing material, if any.
- (6) Remove knurled screw and spacer on each side of board assembly.
- (7) Unplug microprocessor board (rear board of three) from its connector.
- (8) Carefully remove old PROM (located directly under TP1 at top of board) and place it in protective foam.
- (9) Locate pin 1 of new PROM and plug it in with pin 1 nearest C1 on the board. Verify that all pins enter the socket properly.
- (10) For reassembly, follow steps 1-7 in reverse order. Use care when plugging in the microprocessor board to prevent damage to the connector. If foam packing material is provided, reinstall it at the end of the printed circuit board assembly.

After installing the PPOM, verify its operation with the following procedure.

- (1) Connect HP-97 and charger.
- (2) When the power is turned on, the readout light will turn on and the following test message will be loaded into the HP-97 storage registers R1-R4.

R1	=	543210	= time
R2	=	9876543210	= totals
R3	=	9876543210	= R+A
R4	=	9876543210	= A

The readout light will then turn off. Ignore the ERROR message displayed on the HP-97 calculator.

II. DIFFERENCE BETWEEN OLD AND NEW PROM: FRONT-PANEL CONTROL MODE

- (1) The test message to the RS-232 serial port has been eliminated.
- (2) The test message to the HP-97 calculator has been changed to include all digits (0-9).
- (3) Column headings for the data readout have been removed.
- (4) The commas between groups of three digits in data readout have been replaced by spaces.
- (5) A CL X is sent to the HP-97 calculator before data are sent.
- (6) The manual/recycle switch on the front panel is tested at the end of the data readout, allowing the operator to terminate the recycle mode during readout.
- (7) If the electronics package is connected to a terminal either directly or indirectly through a computer, it will echo back any characters received from the terminal or computer while it is operating under the front-panel control mode. Exceptions are:
 - (a) A is not echoed but will transfer control to computer mode.
 - (b) CTRL T is not echoed but will transfer control to terminal mode.
 - (c) CR or LF is echoed as both; that is, CRLF.
 - (d) Characters received during data readout are not echoed.

III. RS-232 SERIAL INTERFACE REQUIREMENTS

Full duplex

Baud rate: 300

Data format: 7 bits, even parity, 1 stop bit

Electronics package rear-panel connector: 25-pin "D" type DBM - 25S

Pin connections:

- 2 - Receive data from computer or terminal
- 3 - Transmit data to computer or terminal
- 4 - 8 (4 and 8 are tied together)
- 7 - Data common
- All other pins: no connection

IV. EXAMPLE OF COMPUTER INTERFACE

Shown below is an example of a Digital Equipment Corp. (DEC) DLV11-J serial interface board configuration for use with LSI-11 computers. This is a 4-channel serial port board. Port 3 is configured for the console terminal; port 1 is configured to interface with the electronics package. (Refer to DEC publication "Memorys and Peripherals"* for more details; see page 2-149 for port 1 jumper locations).

Baud rate selection: port 1 to T = 300 baud

Data parameters:

- D x to 0 = 7 bits
- S x to 0 = 1 stop bit
- P x to 0 = parity enabled
- E x to 1 = even parity

Figure A-1 shows the cable connections to the electronics package and the cable numbers of the Berg connector as viewed from the end of the board.

Fig. A-1.

- (a) Cable between DLV11-J interface board and electronics.
- (b) View of Berg connector from end of board.

*Digital Equipment Corp., Maynard, MA 01574

1-k PROM FOR REMOTE CONTROL: FUNCTIONAL BLOCK DIAGRAM

V 2.1 12 DEC 80
DISK FILENAME: RG7K11

CNTRL

THE POWER TURN ON,
THE REAR PANEL RESET PUSH BUTTON,
AND THE F COMMAND
RETURN CONTROL TO THIS LOOP.

NOTE: * NOT A SUBROUTINE

STOP, CLR, START, LPIA SUBROUTINES:

RDOUT
SERIAL PORT DATA READOUT
FOR COMPUTER OR TERMINAL CONTROL

INT FLG
SETS INTERRUPT FLAG
WHEN IN COMPUTER OR
TERMINAL CONTROL

RON:

ROF:

HPOUT
DATA READOUT TO HP97
UNDER COMPUTER OR TERMINAL CONTROL

PSTAT
CAUSES PRINTING OF
8-CHARACTER STATUS WORD
WHEN REQUESTED BY X COMMAND

STATR

LOADS STATUS BITS INTO
A REGISTER FORMED BY
RAM LOCATIONS
1E (HEX) THROUGH 25

INTST

LOADS ACIA STATUS REGISTER
& PREPARES IT FOR TESTING
FOR INPUT FROM SERIAL PORT

COMP, TMODE (SEE NOTES BELOW)

THIS IS THE REMOTE CONTROL LOOP.
BOTH COMPUTER CONTROL & TERMINAL
CONTROL SHARE THIS LOOP.
TFLAG IS THE CONTROL SWITCH.

STAT:

TRANSMITS STATUS TO
REMOTE CONTROL DEVICE.

CLRF:

CLEARs INTERRUPT FLAG
(IFLG) AND
STATUS CHANGE FLAG (STFLG).

CRLF:

SENDS A CARRIAGE RETURN
AND LINE FEED.

LOAD

TAKES DATA FROM SCALERS
& LOADS INTO MEMORY.

TWO BCD DIGITS PER WORD
BEGINNING AT RAM LOCATION
30 AND ENDING AT LOCATION
42 (HEX).

SUBR
SUBTRACT (SUBTR)

READ
USED BY LOAD SUBROUTINE
TO READ DATA FROM SCALERS.

PRINT
FORMATS & CONTROLS
SCALER DATA READOUT
TO RS232 DEVICE

TYPE:

UNPACKS DIGITS FROM MEMORY,
PERFORMS LEADING ZERO SUPPRESSION,
AND SEPARATES SCALAR DATA
INTO 3-DIGIT GROUPS
SEPARATED BY SPACES.

SEND:

CONVERTS CHARACTERS TO ASCII AND
SENDS THEM THROUGH ACIA TO REMOTE DEVICE
WHEN READY.
ALSO GENERATES CARRIAGE RETURN DELAY.

TEST:

LOADS TEST MESSAGE INTO MEMORY,
AND THEN SENDS IT TO HP97.

HP-97:

PREPARES AND SENDS SCALAR
DATA TO HP97 CALCULATOR
THROUGH HP97 INTERFACE.

VNO:

SENDS FROM VERSION #
TO REMOTE RS232 DEVICE.

APPENDIX C
ASSEMBLY LANGUAGE LISTING

00001				NAM	HLNOC	*12-18-88 NEW CONTROL
00002				OPT	0	
00003				OPT	5	
00004	5000	A	PIAI	EQU	\$5000	*PIA A DATA REGISTER ADDRE
00005	5002	A	PIAO	EQU	\$5002	*PIA B DATA REGISTER ADDRE
00006	5001	A	PIACA	EQU	\$5001	*PIA A CONTROL REG. ADDR.
00007	5003	A	PIACB	EQU	\$5003	*PIA B CONTROL REG. ADDR.
00008	4000	A	ACIAS	EQU	\$4000	*ACIA STATUS ADDRESS
00009	4001	A	ACIAT	EQU	\$4001	*ACIA TRANSMIT ADDRESS
00010	4001	A	ACIAR	EQU	\$4001	*ACIA RECEIVE ADDRESS
00011	007F	A	STACK	EQU	\$7F	
00012	000B	A	RAMB	EQU	\$B	*BEGIN RANDOM ACCESS MEMORY
00013				*		
00014				*****		
00015				*		
00016A	000B			ORG	RAMB	*BEGIN RAM LOCATIONS
00017A	000B	0001	A DLY	RMB	1	
00018A	000C	0001	A VAL	RMB	1	
00019A	000D	0001	A SIGN	RMB	1	*CR+AD)-A SIGN
00020A	000E	0001	A NUM	RMB	1	
00021A	000F	0001	A COM	RMB	1	
00022A	0010	0002	A F	RMB	2	
00023A	0012	0001	A J	RMB	1	
00024A	0013	0001	A K	RMB	1	
00025A	0014	0001	A P	RMB	1	
00026A	0015	0001	A DEL1	RMB	1	*CR DELAY
00027A	0016	0001	A DEL2	RMB	1	
00028A	0017	0002	A R	RMB	2	
00029A	0019	0001	A T	RMB	1	
00030A	001A	0001	A N	RMB	1	
00031A	001B	0001	A C	RMB	1	
00032A	001C	0001	A M	RMB	1	
00033A	001D	0001	A TEN	RMB	1	
00034A	001E	0001	A RUN	RMB	1	*RUN=0/STOP=1 STATUS BIT
00035A	001F	0001	A TOUT	RMB	1	*TIME OUT=1
00036A	0020	0001	A FAULT	RMB	1	*FAULT=1
00037A	0021	0001	A STOPB	RMB	1	*STOP P.B. 1 IF DEPRESSED
00038A	0022	0001	A RESPB	RMB	1	*RESET P.B. 1 IF DEPRESSED
00039A	0023	0001	A STAPB	RMB	1	*START P.B. 1 IF DEPRESSED
00040A	0024	0001	A RCYC	RMB	1	*MANUAL=1/RECYCLE=0
00041A	0025	0001	A ROPB	RMB	1	*READOUT P.B. 1 IF DEPRESS
00042A	0026	0001	A STFLG	RMB	1	*STATUS CHANGE FLAG =1
00043A	0027	0001	A RCYCF	RMB	1	*RECYCLE FLAG *
00044A	0028	0001	A TFLAG	RMB	1	*TERMINAL MODE =1/COMP MOD
00045A	0029	0001	A IFLG	RMB	1	*STATUS INTERRUPT FLAG
00046A	002A	0001	A CHAR	RMB	1	*STORE A ACUM TEMPORARILY
00047		0030	A RAM	EQU	\$30	
00048		0038	A SST	EQU	\$38	
00049				*		
00050				*****		
00051				*		
00052A	73FE			ORG	\$73FE	*RESTART ADDRESS
00053A	73FE	7028	A RSTRT	FDB	CNTRL	*POINTER TO CONTROL PROGRA
00054				*		
00055				*****		
00056				*		
00057A	7000			ORG	\$7000	
00058A	7000	53	A VTBL	FDB	/S/	


```

00059A 7001      70A1  A      FDB      START      *VECTORS FOR COMAND CHAR'S
00060A 7003      48      A      FCC      /H/
00061A 7004      7099  A      FDB      STOP
00062A 7006      5A      A      FCC      /Z/
00063A 7007      709D  A      FDB      CLR
00064A 7009      52      A      FCC      /R/
00065A 700A      70BC  A      FDB      RDOUT      *READOUT VIA RS232
00066A 700C      50      A      FCC      /P/
00067A 700D      7ACE  A      FDB      HPDOUT     *READOUT VIA HPS7
00068A 700F      58      A      FCC      /X/
00069A 7010      70F3  A      FDB      PSTAT     *REQUEST STATUS
00070A 7012      49      A      FCC      /I/
00071A 7013      70E9  A      FDB      INTPLG     *SET STATUS INTERRUPT FLAG
00072
00073          *****
00074          *CHRRIDGE RETURN = CR
00075          *LINE FEED = LF
00076A 7015      000A  A  S1      FDB      $000A      *CR, LF
00077A 7017      56      A      FCC      L3.W 2.1 12 DEC 80*VERSION NO.
00078A 7026      000A  A      FDB      $000A      *CR, LF*
00079          *****
00080A 7028 0F          CNTRL SEI          *DISABLE INTERRUPT
00081A 7029 8E 007F  A      LDS      #STACK     *LOAD STACK POINTER
00082A 702C BD 71C1  A      JSR      INIT      *JUMP TO INITIALIZE ROUTIN
00083A 702F 3D 68 7099 BSR      STOP      *STOP SR
00084A 7031 8D 6A 709D BSR      CLR      *CLEAR SR
00085A 7033 BD 7090  A      JSR      RON      *TURN ON READOUT LIGHT
00086A 7036 BD 73AE  A      JSR      TEST     *JUMP TO TEST SUBROUTINE
00087A 7039 BD 70B6  A      JSR      ROF      *TURN OFF READOUT LIGHT
00088A 703C BD 70F9  A  CNTRL1 JSR      STATR     *READ/STORE STATUS
00089A 703F 86 01      A      LDAA     #1        *SET T FLAG
00090A 7041 97 28      A      STAA     TFLAG
00091A 7043 BD 7116  A      JSR      INTST     *IS THERE AN INPUT
00092A 7046 24 1D 7065 BCC      TERM3     *NO INPUT, BRANCH
00093A 7048 B6 4001  A      LDAA     ACIAR
00094A 704B 81 5E      A      CMPA     #$5E      *IS IT A "C" ?
00095A 704D 26 03 7052 BNE      TERM0     *NO , BRANCH
00096A 704F 7E 711C  A      JMP      CCMP     *YES, JMP TO COMPUTER CONT
00097A 7052 81 14      A  TERM0  CMPA     #$14     *IS IT CONTROL T
00098A 7054 26 03 7059 BNE      TERM1     *IF YES JMP TO TERM MODE
00099A 7056 7E 711F  A      JMP      TMODE
00100A 7059 81 0E      A  TERM1  CMPA     #$0E
00101A 705B 2E 05 7062 BGT      TERM2     *ECHO IT
00102A 705D BD 71B2  A      JSR      CRLF     *ECHO CR AND LF
00103A 7060 20 03 7065 BRA      TERM3
00104A 7062 BD 7326  A  TERM2  JSR      SEND
00105A 7065 96 1E      A  TERM3  LDAA     RUN      *BRANCH IF STOPPED
00106A 7067 26 06 706F BNE      CNTRL2     *BRANCH IF STOPPED
00107A 7069 96 24      A      LDAA     RCYC      *RECYCLE STATUS
00108A 706B 97 27      A      STAA     RCYCF     *RECYCLE FLAG
00109A 706D 20 CD 703C BRA      CNTRL1
00110A 706F 96 1F      A  CNTRL2 LDAA     TOUT     *TEST FOR TIME OUT
00111A 7071 27 04 7077 BEQ      CNTRL3     *BRANCH IF NO TIME OUT
00112A 7073 96 27      A      LDAA     RCYCF     *RECYCLE MODE ?
00113A 7075 27 04 707B BEQ      CNTRL4     *BRANCH IF IN RECYCLE
00114A 7077 96 25      A  CNTRL3 LDAA     ROPB     *IS R. O. BUTTON DEPRESSED
00115A 7079 27 C1 703C BEQ      CNTRL1     *BRANCH IF NOT
00116A 707B 8D 33 70B0 CNTRL4 BSR      RON      *TURN ON R. O. LIGHT

```

```

00117A 707D 8D 71DE A JSR LOAD *READ THE SCALERS
00118A 7080 8D 734B A JSR HP97 *SEND RESULTS TO HP-97
00119A 7083 8D 7221 A JSR SUBTR *SUBTRACT A FROM R+A
00120A 7086 8D 726C A JSR PRINT *PRINT RESULTS
00121A 7089 8D 2B 70B6 BSR ROF *TURN OFF READOUT LIGHT
00122A 708B 8D 6C 70F9 BSR STATR
00123A 708D 96 24 A LDAA RLYC
00124A 708F 97 27 A STAA RLYC
00125A 7091 26 A9 703C ONE UNTPL1 *BRANCH IF MANUAL
00126A 7093 8D 08 709D BSR CLR *CLEAR SR
00127A 7095 8D 0A 70A1 BSR START *START SR
00128A 7097 2A A3 703C BPH UNTPL1 *RETURN TO WAIT LOOP
00129 ****
00130A 7099 06 28 A STOP LDAA #0 *DATA STOP CODE
00131A 709B 2A 06 70A7 BRA CLR
00132A 709D 06 29 A CLR LDAA #0 *DATA CLEAR CODE
00133A 709F 2A 02 70A1 BRA LTR
00134A 70A1 06 2A A START LDAA #0 *DATA START CODE
00135A 70A3 F7 5002 A LTR STAA #0 *SEND DATA TO SCALERS CODE
00136A 70A6 86 5000 A LDAA #0
00137A 70A9 86 5000 A LDAA #0
00138A 70AC 7F 5002 A CLR *END OF CONTROL
00139A 70AF 19 RTS *RESTART
00140 ****
00141A 70B0 86 3C A RUN LDAA #0 *LOAD CONTROL REG CODE
00142A 70B2 B7 5003 A STAA #0 *SEND IT OUT
00143A 70B5 39 RTS *RETURN
00144A 70B6 86 34 A ROF LDAA #0 *LOAD CONTROL REG CODE
00145A 70B8 B7 5001 A STAA #0 *SEND IT OUT
00146A 70BB 39 RTS *RETURN
00147A 70BC 96 1E A RDOUT LDAA #0 *IS IT RUNNING?
00148A 70BE 27 A0 70C0 BSR RDOUT *SEND IF YES
00149A 70C0 8D 0E 70C0 BSR ROF *SEND OUT TO MACHINE
00150A 70C2 8D 71DE A JSR LOAD
00151A 70C5 8D 7221 A JSR SUBTR
00152A 70C8 8D 726C A JSR PRINT
00153A 70CB 8D E9 70B6 BSR ROF
00154A 70CD 39 RDOUT1 RTS
00155 ****
00156A 70CE 96 1E A HPDUT LDAA #0 *HP97 OUTPUT ROUTINE
00157A 70D0 27 16 70E8 BSR HPDUT1
00158A 70D2 8D 0C 70B0 BSR ROF
00159A 70D4 8D 71DE A JSR LOAD
00160A 70D7 8D 734B A JSR HP97
00161A 70DA 8D 0A 70A1 BSR ROF
00162A 70DC 96 29 A LDAA IFLG *IS STATUS INTERRUPT FLAG S
00163A 70DE 27 08 70E8 BSR HPDUT1 *IF NOT BRANCH
00164A 70E0 96 26 A LDAA STFLG *PRINT STATUS FLAG
00165A 70E2 8D 7326 A JSR SEND
00166A 70E5 8D 71AD A JSR CLR
00167A 70E8 39 HPDUT1 RTS
00168 ****
00169A 70E9 86 01 A INTFLG LDAA #1
00170A 70EB 97 29 A STAA IFLG
00171A 70ED 8D 0A 70F9 BSR STATR
00172A 70EF 7F 0026 A CLR STFLG
00173A 70F2 39 RTS
00174 ****

```

PAGE 004 PGT 11 SA 1 HEND

[illegible]

```

00233A 715C 8D 7126 A JSR SEND *PRINT IT
00234A 715F 8D 7182 A JSR CRLF *ECHO CF AND LF
00235A 7162 20 09 716D BRA COMF6
00236A 7164 8D 4C 7182 COMF4 BSR CRLF
00237A 7166 20 8C 7124 BRA COMF2
00238A 7168 8D 7126 A COMF5 JSR SEND
00239A 716B 20 87 7124 BRA COMF2
00240A 716D 81 46 A COMF6 CMPA #$46 *IS IT A "F"?
00241A 716F 26 07 7178 BNE VECT *IF NO BRANCH TO VECT
00242A 7171 86 01 A LDAA #1
00243A 7173 97 27 A STAA RCVCF
00244A 7175 7E 701C A JMP CNTRL1 *IF YES RETURN TO MANUAL C
00245 ***
00246A 7178 0E 0000 A VECT1 LDX #VECT1 *LOAD VECTOR POINTER
00247A 717B A1 00 A VECT1 CMPA 0,X
00248A 717D 27 1A 7199 BEQ VECT0 *IF YES LOAD VECTOR
00249A 717F 08 INX *IF NO BUMP POINTER
00250A 7180 08 INX
00251A 7181 08 INX
00252A 7182 8C 7015 A CPX #VECT1+21 *LAST ONE ?
00253A 7185 26 F4 7178 BNE VECT1
00254A 7187 96 28 A LDAA TFLAG
00255A 7189 27 0C 7197 BEQ VECT2 *IN COMPUTER MODE DON'T
00256 *ACKNOWLEDGE ILLEGAL CHARACTERS
00257A 718B 86 07 A LDAA #$07 *ASCII "BELL"
00258A 718D 8D 712C A JSR SEIP1 *SEND BELL CODE
00259A 7190 86 1F A LDAA #$1F *IF NOT VALID CHARACTER
00260A 7192 8D 7126 A JSR SEND *PRINT " "
00261A 7195 8D 18 7182 BSR CRLF *PRINT CRLF
00262A 7197 20 8B 7124 VECT1 BRA COMF2 *RETURN TO WAITING LOOP
00263A 7199 EE 01 A VECT1 LDX 1,X
00264A 719B AD 00 A JSR 0,X *JUMP TO VECTORED ROUTINE
00265A 719D 20 F8 7197 BRA VECT2
00266 ***
00267A 719F 0E 001E A STAT1 LDX #RUN *TRANSMIT STATUS
00268A 71A2 A6 00 A STAT1 LDAA 0,X
00269A 71A4 8D 7126 A JSR SEND
00270A 71A7 08 INX
00271A 71A8 8C 0026 A CPX #RUN+8
00272A 71AB 26 F5 71A2 BNE STAT11
00273A 71AD 4F CLRF CLRA
00274A 71AE 97 29 A STAA IFLG
00275A 71B0 97 26 A STAA STFLG
00276A 71B2 97 2A A CRLF STAA CHAR *STORE A ACUM TEMPORARILY
00277A 71B4 86 00 A LDAA #$00
00278A 71B6 8D 7126 A JSR SEND
00279A 71B9 86 0A A LDAA #$0A
00280A 71BB 8D 7126 A JSR SEND
00281A 71BE 96 2A A LDAA CHAR *RESTORE A ACUM
00282A 71C0 39 RTS
00283 *
00284 *****
00285 *
00286A 71C1 4F INIT CLRA *CLEAR ACCUMULATOR A
00287A 71C2 97 26 A STAA STFLG *CLEAR STATUS FLAG
00288A 71C4 B7 5000 A STAA PIAI *DEFINE A SIDE AS INPUT
00289A 71C7 43 COMA
00290A 71C8 B7 5002 A STAA PIAO *DEFINE B SIDE AS OUTPUT

```

PAGE 006 RG7K11 SA-1 HLNCC

```

00291A 710B 86 04 A LDAA #34
00292A 710D 87 5001 A STAA PIACH *CHANGE FROM DATA DIRECTIO
00293 * REGISTER TO DATA REGISTER
00294
00295A 710D 87 5003 A STAA PIACB *DITTO
00296A 710D 86 03 A LDAA #3 *MASTER RESET
00297A 7105 B7 400A A STAA ACIAS
00298A 7108 86 09 A LDAA #39 *SET UP DATA FORMAT
00299A 710A B7 4000 A STAA ACIAS
00300A 710D 39 RTS
00301 *
00302 *****
00303 *
00304A 710E 0E 0030 A LOAD LDX #RHM *LOAD INDEX REGISTER WITH
00305 * FIRST STORAGE LOCATION
00306A 71E1 86 01 A LDAA #1 *LOAD CONTROL CODE "1" INT
00307 * PIA B DATA REGISTER
00308A 71E3 B7 5002 A STAA PIA0
00309A 71E6 8D 1E 7206 LLP1 BSR READ *READ DIGIT. STORE IN VAL
00310A 71E8 06 0C A LDAB VAL *MOVE DIGIT LEFT
00311A 71EA 58 ASLB
00312A 71EB 58 ASLB
00313A 71EC 58 ASLB
00314A 71ED 58 ASLB
00315A 71EE 7C 5002 A INC PIA0 *INCREMENT CONTROL CODE
00316 * FOR NEXT DIGIT
00317A 71F1 8D 1E 7206 BSR READ *READ THE DIGIT
00318A 71F3 0A 0C A ORAB VAL *FORM A 2-DIGIT PAIR
00319A 71F5 E7 00 A STAB 000 *STORE RESULT AT POINTER
00320A 71F7 08 INX *INCREMENT POINTER
00321A 71F8 8C 0042 A CPX #RAM+18 *POINTER=42?
00322A 71FB 27 05 7202 BEO LJP1 *YES, GO TO LJP1
00323A 71FD 7C 5002 A INC PIA0 *NO, INCREMENT CONTROL COD
00324A 7200 20 E4 71E6 BRA LLP1 *BRANCH TO READ NEXT
00325 * PAIR OF DIGITS
00326A 7202 7F 5002 A LJP1 CLR PIA0 *REMOVE CONTROL CODE
00327A 7205 39 RTS *RETURN FROM SUBROUTINE
00328A 7206 B6 5000 A READ LDAA PIA1 *LOAD THE PIA A DATA REGIS
00329A 7209 2B FB 7206 BMJ READ *PROCEED IF POSITIVE
00330 * (DATA PRESENT)
00331A 720B 86 FF A LDAA #FF *ALLOW DELAY FOR
00332 * SETTLING TIME
00333A 720D 97 0B A STAA DLY
00334A 720F 7A 000B A LLP2 DEC DLY
00335A 7212 26 FB 720F BNE LLP2
00336A 7214 B6 5000 A LDAA PIA1 *LOAD THE PIA A DATA
00337 * REGISTER REDUNDANTLY
00338A 7217 84 0F A ANDA #F *STRIP OFF LEAST
00339 * SIGNIFICANT 4 BITS
00340A 7219 81 0F A CMPA #F *DATA=F?
00341A 721B 26 01 721E BNE LJP2 *NO, PROCEED
00342A 721D 4F CLRA *YES, CHANGE TO ZERO
00343A 721E 97 0C A LJP2 STAA VAL *STORE RESULT IN VAL
00344A 7220 39 RTS *RETURN FROM SUBROUTINE
00345 *
00346 *****
00347 *
00348A 7221 7F 000D A SUBTR CLR SIGN *CLEAR SIGN CONTROL

```

00349				*				CONSTANT (0 FOR +; 1 FOR
00350A	7224	0E	0038	A	LDX	#SST		*LOAD STARTING ADDRESS
00351A	7227	06	05	A	LDAB	#5		*LOAD CONTROL CONSTANT
00352A	7229	86	99	A	SLP1	LDAA	#99	*LOAD \$99
00353A	722B	A0	09	A	SUBA	\$0,X		*SUBTRACT TO COMPLEMENT
00354A	722D	A7	0E	A	STAA	\$E,X		*STORE COMPLEMENT
00355A	722F	09			DEX			*DECREMENT ADDRESS
00356A	7230	5A			DECB			*DECREMENT CONTROL NUMBER
00357A	7231	26	F6	7229	BNE	SLP1		*RECYCLE UNTIL ALL DIGITS
00358				*				ARE COMPLEMENTED
00359A	7233	00			SEC			*SET THE CARRY
00360A	7234	06	05	A	LDAB	#5		*LOAD CONTROL NUMBER
00361A	7236	0E	0038	A	LDX	#SST		*LOAD STARTING ADDRESS
00362A	7239	A6	04	A	SLP2	LDAA	4,X	*LOAD PAIR OF DIGITS
00363				*				FOR ADDITION
00364A	723B	A9	0E	A	ADCA	\$E,X		*ADD WITH CARRY
00365A	723D	19			DAA			*DECIMAL ADJUST
00366A	723E	A7	13	A	STAA	\$13,X		*STORE SUM
00367A	7240	09			DEX			*DECREMENT ADDRESS
00368A	7241	5A			DECB			*DECREMENT CONTROL NUMBER
00369A	7242	26	F5	7239	BNE	SLP2		*RECYCLE UNTIL ADDITION
00370				*				IS COMPLETE
00371A	7244	24	01	7247	BCC	SJP1		*BRANCH TO SJP1 IF
00372				*				CARRY IS CLEAR
00373A	7246	39			RTS			*RETURN TO CONTROL PROGRAM
00374				*				IF CARRY IS SET
00375A	7247	86	01	A	SJP1	LDAA	#1	*CHANGE SIGN TO -
00376A	7249	97	00	A	STAA	SIGN		
00377A	724B	0E	0038	A	LDX	#SST		*REVERSE SUBTRAHEND AND
00378				*				MINUEND AND REPEAT
00379A	724E	06	05	A	LDAB	#5		
00380A	7250	86	99	A	SLP4	LDAA	#99	
00381A	7252	A0	13	A	SUBA	\$13,X		
00382A	7254	A7	13	A	STAA	\$13,X		
00383A	7256	09			DEX			
00384A	7257	5A			DECB			
00385A	7258	26	F6	7250	BNE	SLP4		
00386A	725A	00			SEC			
00387A	725B	06	05	A	LDAB	#5		
00388A	725D	0E	0038	A	LDX	#SST		
00389A	7260	86	00	A	SLP5	LDAA	#0	
00390A	7262	A9	13	A	ADCA	\$13,X		
00391A	7264	19			DAA			
00392A	7265	A7	13	A	STAA	\$13,X		
00393A	7267	09			DEX			
00394A	7268	5A			DECB			
00395A	7269	26	F5	7260	BNE	SLP5		
00396A	726B	39			RTS			
00397				*				
00398				*****				
00399				*				
00400A	726C	86	05	A	PRINT	LDAA	#5	*SET NUM=5
00401A	726E	97	0E	A	STAA	NUM		
00402A	7270	86	30	A	LDAA	#RAM		*LOAD STARTING ADDRESS
00403				*				(\$0030) AT F, F+1
00404A	7272	97	11	A	STAA	F+1		
00405A	7274	4F			CLRA			
00406A	7275	97	10	A	STAA	F		

```

00407A 7277 97 0F      A      STAR  COM      *SET COM=0
00408A 7278 8D 4F 720A  BSR  TYPE  *TYPE OUT TIME
00409A 727B 8D 40 72BD  BSR  SP      *PRINT 3 SPACES
00410A 727D 86 09      A      LDAA  ##9      *SET NUM=9
00411A 727F 97 0E      A      STAR  NUM
00412A 7281 86 33      A      LDAA  #RAM+0  *SET START ADDR = #0033
00413A 7283 97 11      A      STAR  F+1
00414A 7285 86 02      A      LDAA  ##2      *SET COM=2
00415A 7287 97 0F      A      STAR  COM
00416A 7289 8D 3F 720A  BSR  TYPE  *TYPE OUT TOTALS
00417A 728B 8D 30 72BD  BSR  SP      *PRINT 3 SPACES
00418A 728D 86 38      A      LDAA  #RAM+0  *SET START ADDR = #0038
00419A 728F 97 11      A      STAR  F+1
00420A 7291 8D 37 720A  BSR  TYPE  *TYPE OUT M+A
00421A 7293 8D 38 72BD  BSR  SP      *PRINT 3 SPACES
00422A 7295 86 3D      A      LDAA  #RAM+0  *SET START ADDR = #004D
00423A 7297 97 11      A      STAR  F+1
00424A 7299 8D 2F 720A  BSR  TYPE  *TYPE OUT ACCIDENTALS
00425A 729B 86 28      A      LDAA  FLAG  *PRINT (F+A)+H IF TERM MOD
00426A 729D 27 1A 72B9  BEQ  PRJP1  *DONT PRINT IF COMPUTER MO
00427A 729F 8D 1C 72BD  BSR  SP      *TYPE 3 SPACES
00428A 72A1 86 01      A      LDAA  #1      *IS (R+A)-A POSITIVE?
00429A 72A3 91 0D      A      CNPA  SIGN
00430A 72A5 27 07 72AE  BEQ  PRJP1  *NO. GO TO PRJP1
00431A 72A7 86 2B      A      LDAA  ##2B  *YES. PRINT +
00432A 72A9 8D 7326 A      JSR  SEND
00433A 72AB 20 05 72B3  BRA  PRJP2  *GO TO PRJP2
00434A 72AD 86 2D      A PRJP1 LDAA  ##2D  *PRINT -
00435A 72B0 8D 7326 A      JSR  SEND
00436A 72B3 86 47      A PRJP2 LDAA  #RAM+23 *SET START ADDR = #0047
00437A 72B5 97 11      A      STAR  F+1
00438A 72B7 8D 11 720A  BSR  TYPE  *TYPE OUT (R+A)-A
00439A 72B9 8D 71B2 A PRJP2 JSR  CRLF  *PRINT CARRIAGE RTN, LINE F
00440A 72BC 39          RTS      *RETURN TO CONTROL PROGRAM
00441A 72BD 86 20      A SP  LDAA  ##20  *PRINT 3 SPACES
00442          *      SUBROUTINE
00443A 72BF 8D 65 7326  BSR  SEND
00444A 72C1 86 20      A      LDAA  ##20
00445A 72C3 8D 61 7326  BSR  SEND
00446A 72C5 86 20      A      LDAA  ##20
00447A 72C7 8D 5D 7326  BSR  SEND
00448A 72C9 39          RTS      *RETURN
00449          *
00450          *****
00451          *
00452A 72CA 4F          TYPE  CLRA      *CLEAR CONTROL #'S J,P, &
00453A 72CB 97 12      A      STAR  J
00454A 72CD 97 14      A      STAR  P
00455A 72CF 97 13      A      STAR  K
00456A 72D1 0E 10      A      LOX  F      *LOAD STARTING ADDRESS IN
00457A 72D3 A6 00      A TYLP1 LDAA  0,X  *LOAD A PAIR OF DIGITS
00458A 72D5 06 14      A      LDAB  P      *NEED LEAST SIG. DIGIT?
00459A 72D7 27 04 72D0  BEQ  B1      *NO. GO TO B1
00460A 72D9 84 0F      A      ANDA  ##F  *YES, GET LEAST SIG. 4 BIT
00461A 72DB 20 06 72E3  BRA  B2      *GO TO B2
00462A 72DD 84 F0      A B1  ANDA  ##F0  *PICK OFF MOST SIG. 4 BITS
00463A 72DF 46          RORA  *MOVE RIGHT 4 BITS
00464A 72E0 46          RORA

```

```

00465H 72E1 46          POKA
00466A 72E2 46          ROKA
00467A 72E3 4D          B2          TSTA
00468A 72E4 27 08 72EE  B5          *DIGIT=0?
00469A 72E5 06 01      A          B5          *YES, GO TO B3
00470A 72E6 07 13      A          LDAB      #1          *NO, SET K=1
00471A 72E7 8D 1A 7326  B5          STAB      K
00472A 72E8 20 0F 72FD  B5          B5?      SEND          *PRINT THE CHARACTER
00473A 72E9 71 0013      A          BRA      B4          *GO TO B4
00474A 72FA 26 F7 72EA  B5          TST      K          *K=0?
00475A 72FB 06 0E      A          BNE      B5          *NO, PRINT THE DIGIT
00476          *          LDAB      NUM          *YES, IS THIS THE LAST DIGIT
00477A 72F5 01 12      A          *          IN THE NUMBER?
00478A 72F7 27 F1 72EA  B5          CMPB      J
00479A 72F9 8A 20      A          B5          *YES, PRINT THE ZERO
00480A 72FA 20 ED 72EA  B5          LDAB      ##20          *NO, PRINT A SPACE
00481A 72FD 96 0F      A          BRH      B5
00482          *          LDAB      NUM          *IS THIS THE LAST DIGIT
00483A 72FF 91 17      A          *          IN THE NUMBER?
00484A 7301 26 01 7304  B5          CMPA      J
00485A 7303 39          BNE      B6          *NO, GO TO B6
00486A 7304 7C 000F      A          RTS          *YES, RETURN FROM SUBROUTINE
00487A 7307 86 03      A          INC      COM          *INCREMENT COM
00488A 7309 91 0F      A          LDAB      #3          *COM=3?
00489A 730B 26 07 7314  B5          CMPA      COM
00490A 730D 86 20      A          BNE      B7          *NO, GO TO B7
00491A 730F 8D 15 7326  B5          LDAB      ##20          *PRINT A SPACE
00492A 7311 7F 000F      A          B5?      SEND
00493A 7314 7C 0012      A          CLR      COM          *SET COM=0
00494A 7317 7C 0014      A          INC      J          *INCREMENT J AND P
00495A 731A 86 02      A          INC      P
00496A 731C 91 14      A          LDAB      #2          *P=2?
00497A 731E 26 B3 72D3  B5          CMPA      P
00498A 7320 7F 0014      A          BNE      TYLP1          *NO, GO TO TYLP1
00499A 7323 08          CLR      P          *YES, SET P=0
00500A 7324 20 AD 72D3  B5          INX          *STEP TO NEXT PAIR OF DIGIT
00501          *          BRA      TYLP1
00502          *
00503          *****
00504A 7326 81 09      A          *          SEND          CMPA      #9          *CHARACTER>9?
00505A 7328 2E 02 732C  B5          BGT      SEJP1          *YES, PROCEED
00506A 732A 8A 30      A          ORAA      ##30          *NO, FORM ASCII CODE
00507A 732C F6 4000      A          LDAB      ACIAS          *LOAD STATUS OF ACIA
00508A 732F 56          RORB
00509A 7330 56          RORB
00510A 7331 24 F9 732C  B5          BCC      SEJP1          *PROCEED IF ACIA IS READY
00511A 7333 B7 4001      A          STAB      ACIAT          *SEND OUT CHARACTER
00512          *          TO TERMINAL
00513A 7336 81 0D      A          *          *IS IT A CR?
00514A 7338 26 10 734A  B5          CMPA      ##0D
00515A 733A 06 5F      A          BNE      DL2
00516A 733C 07 15      A          LDAB      #5F          *IF YES START A DELAY
00517A 733E 07 16      A          STAB      DEL1
00518A 7340 7A 0015      A          STAB      DEL2
00519A 7343 26 FB 7340  B5          DEC      DEL1          *BEGIN DELAY
00520A 7345 7A 0016      A          BNE      DL1
00521A 7348 26 F6 7340  B5          DEC      DEL2
00522A 734A 39          BNE      DL1          *DELAY FINISHED?
00523A 734B          *          RTS          *YES, RETURN

```



```

00523      *
00524      *****
00525      *
00526A 734B 86 4D      A HP97      LDAA      #$4D      *HP97 "CLRX"
00527A 734D 8D 4D 739C      BSR      OUT
00528A 734F 86 04      A          LDAA      #4          *SET M=4
00529A 7351 97 1C      A          STAA      M
00530A 7353 86 01      A          LDAA      #1          *SET N=1
00531A 7355 97 1A      A          STAA      N
00532A 7357 86 03      A          LDAA      #3          *SET C=3
00533A 7359 97 1B      A          STAA      C
00534A 735B CE 0030      A          LDX      #RAM      *LOAD STARTING ADDRESS
00535A 735E A6 00      A HPLP1     LDAA      0,X      *LOAD A PAIR OF DIGITS
00536A 7360 8D 25 7387      BSR      PR      *JUMP TO UNPACKING ROUTINE
00537A 7362 08          INX          *INCREMENT THE ADDRESS
00538A 7363 7A 001B      A          DEC      C      *DECREMENT THE CONTROL NUM
00539A 7366 27 02 736A      BEQ      HPJP1
00540A 7368 20 F4 735E      BRA      HPLP1      *LOOP IF NOT ZERO
00541A 736A 86 4F      A HPJP1     LDAA      #$4F      *LOAD STORE CODE
00542A 736C 8D 2E 739C      BSR      OUT      *SEND IT OUT
00543A 736E 96 1A      A          LDAA      N      *FORM CODE FOR STORAGE
00544      *          ADDRESS ("OR" N WITH $40)
00545A 7370 8A 40      A          ORAA      #$40
00546A 7372 8D 28 739C      BSR      OUT      *SEND IT OUT
00547A 7374 7C 001A      A          INC      N      *INCREMENT STORAGE ADDRESS
00548      *          CONTROL NUMBER
00549A 7377 7A 001C      A          DEC      M      *ALL NUMBERS SENT OUT?
00550A 737A 27 06 7382      BEQ      HPJP2      *YES, LEAVE LOOP
00551A 737C 86 05      A          LDAA      #5      *NO, SET C=5 & RECYCLE
00552A 737E 97 1B      A          STAA      C
00553A 7380 20 DC 735E      BRA      HPLP1
00554A 7382 86 4A      A HPJP2     LDAA      #$4A      *LOAD CODE FOR LABEL A
00555A 7384 8D 16 739C      BSR      OUT      *SEND IT OUT
00556A 7386 39          RTS          *RETURN TO CONTROL PROGRAM
00557A 7387 97 1D      A PR      STAA      TEM      *SAVE CONTENTS OF A IN TEM
00558A 7389 84 F0      A          ANDA      #$F0      *PICK OFF MOST SIG. BYTE
00559A 738B 44          LSR          *MOVE IT OVER
00560A 738C 44          LSR
00561A 738D 44          LSR
00562A 738E 44          LSR
00563A 738F 8A 40      A          ORAA      #$40      *FORM CODE FOR HP-97
00564A 7391 8D 09 739C      BSR      OUT      *SEND IT OUT
00565A 7393 96 1D      A          LDAA      TEM      *RECOVER DATA
00566A 7395 84 0F      A          ANDA      #$0F      *PICK OFF LEAST SIG. BYTE
00567A 7397 8A 40      A          ORAA      #$40      *FORM PROPER CODE
00568A 7399 8D 01 739C      BSR      OUT      *SEND IT OUT
00569A 739B 39          RTS          *RETURN
00570A 739C B7 5002      A OUT     STAA      PIA0      *SEND CODE TO HP-97
00571      *          INTERFACE
00572A 739F C6 19      A          LDAB      #19      *LOAD DELAY ARGUMENT
00573A 73A1 BD 733C      A          JSR      DL0      *GO TO DELAY LOOP
00574A 73A4 4F          CLRA          *LOAD BLANK CODE TO
00575      *          RELEASE HP-97
00576A 73A5 B7 5002      A          STAA      PIA0
00577A 73A8 C6 20      A          LDAB      #20
00578A 73AA BD 733C      A          JSR      DL0      *GO TO DELAY LOOP
00579A 73AD 39          RTS
00580      *

```

```

00581
00582
00583A 73AE CE 0041 A TEST LDX #RAM+17 *LOAD STARTING ADDRESS
00584A 73B1 0C TEST1 CLC *CLEAR CARRY
00585A 73B2 86 10 A LDAA ##10 *DECIMAL "10"
00586A 73B4 A7 00 A TEST2 STAA 0,X
00587A 73B6 8B 22 A ADDA ##22 *DECIMAL "22"
00588A 73B8 19 DAA *CONVERT SUM TO DECIMAL
00589A 73B9 09 DEX *DECREMENT POINTER
00590A 73BA 8C 002F A CPX #RAM-1 *LAST LOCATION
00591A 73BD 27 04 73C3 BEQ TEST3
00592A 73BF 25 F0 73B1 BCS TEST1 *START SEQUENCE 0123456 ET
00593A 73C1 20 F1 73B4 BRA TEST2
00594A 73C3 BD 734B A TEST3 JSR HP97 *JUMP TO HP-97 OUTPUT ROUT
00595A 73C6 39 RTS
00596
00597A 73C7 CE 7015 A VNO LDX #S1 *LOAD STARTING ADDRESS
00598A 73CA A6 00 A TELP1 LDAA 0,X *LOAD VERSION NUMBER
00599A 73CC BD 7326 A JSR SEND *PRINT IT
00600A 73CF 08 INX *INCREMENT ADDRESS
00601A 73D0 8C 7028 A CPX #S1+19 *HAS LAST CHAR. BEEN PRINT
00602A 73D3 26 F5 73CA BNE TELP1 *NO. GET NEXT CHARACTER
00603A 73D5 39 RTS *YES. RETURN
00604
END
TOTAL ERRORS 00000---00000

```

```

4001 ACIAR 00010*00093 00218 00219
4000 ACIAS 00008*00198 00297 00299 00507
4001 ACIAT 00009*00511
7200 B1 00459 00462*
72E3 B2 00461 00467*
72EE B3 00468 00473*
72FD B4 00472 00481*
72EA B5 00471*00474 00478 00480
7304 B6 00484 00486*
7314 B7 00489 00493*
7311 B9 00492*
0018 C 00031*00533 00538 00552
711C CCMP 00096 00204*
7122 CCMP1 00206*
7124 CCMP2 00207*00216 00237 00239 00262
7135 CCMP3 00208 00211 00215*
7164 CCMP4 00225 00227 00236*
7168 CCMP5 00223 00238*
716D CCMP6 00221 00235 00240*
002A CHAR 00046*00228 00231 00276 00281
709D CLR 00063 00084 00126 00132*
71AD CLRf 00166 00214 00273*
7028 CNTRL 00053 00080*
703C CNTRL1 00088*00109 00115 00125 00128 00244
706F CNTRL2 00106 00110*
7077 CNTRL3 00111 00114*
707B CNTRL4 00113 00116*
000F COM 00021*00407 00415 00486 00488 00492
7182 CRLF 00102 00234 00236 00261 00276*00439
0015 DEL1 00026*00516 00518

```

PAGE 012 RG7K11 SR:1 HLNCC

```

0016 DEL2 00027*00517 00520
7330 DL0 00516*00573 00578
7340 DL1 00518*00519 00521
734A DL2 00514 00522*
000B DLY 00017*00333 00334
0010 F 00022*00404 00406 00413 00419 00423 00437 00456
0020 FAULT 00036*
734B HP97 00118 00160 00526*00594
736A HPJP1 00539 00541*
7382 HPJP2 00550 00554*
735E HPLP1 00535*00540 00553
700E HPOUT 00067 00156*
70E8 HPOUT1 00157 00163 00167*
0029 IFLG 00045*00162 00170 00207 00274
7101 INIT 00082 00286*
70E9 INTFLG 00071 00163*
7116 INTST 00091 00197*00215
0012 J 00023*00453 00477 00483 00493
0013 K 00024*00455 00470 00473
7202 LJP1 00322 00326*
721E LJP2 00341 00343*
71E6 LLP1 00309*00324
720F LLP2 00334*00335
710E LOAD 00117 00150 00154 00304*
70A3 LPIA 00131 00133 00135*00181
0010 M 00032*00529 00549
001A N 00030*00531 00543 00547
000E NUM 00020*00401 00411 00475 00481
739C OUT 00527 00542 00546 00555 00564 00568 00570*
0014 P 00025*00454 00458 00494 00496 00498
5001 PIACA 00006*00292
5003 PIACB 00007*00142 00145 00295
5000 PIAI 00004*00136 00137 00288 00328 00336
5002 PIAO 00005*00135 00138 00290 00308 00315 00323 00326 00570
00576
7387 PR 00536 00557*
726C PRINT 00120 00152 00400*
72AE PRJP1 00430 00434*
72B3 PRJP2 00433 00436*
72B9 PRJP3 00426 00439*
70F3 PSTAT 00069 00175*
0017 R 00028*
0030 RAM 00047*00304 00321 00402 00412 00418 00422 00436 00534
00583 00590
000B RAMB 00012*00016
0024 RCYC 00040*00107 00123
0027 RCYCF 00043*00108 00112 00124 00243
70BC RDOUT 00065 00147*
70CD RDOUT1 00148 00154*
7206 READ 00309 00317 00328*00329
0022 RESPB 00038*
70B6 ROF 00087 00121 00144*00153 00161
70B0 RON 00085 00116 00141*00149 00158
0025 ROPB 00041*00114
73FE RSTRT 00053*
001E RUN 00034*00105 00147 00156 00179 00267 00271
7015 S1 00075*00597 00601
732C SEJP1 00258 00505 00507*00510

```

```

7326 SEND      00104 00165 00212 00230 00233 00238 00260 00269 00278
               00280 00432 00435 00443 00445 00447 00471 00491 00504*
               00599
0000 SIGN      00019*000348 00376 00429
7247 SJP1      00371 00375*
7229 SLP1      00352*00357
7239 SLP2      00362*00369
7250 SLP4      00380*00385
7260 SLP5      00389*00395
7280 SP        00409 00417 00421 00427 00441*
0038 SST       00048*00350 00361 00377 00388
007F STACK     00011*00081
0023 STAPB     00039*
70A1 START     00059 00137 00134*
70F9 STATR     00088 00122 00171 00175 00179*00209 00213 00217
70FE STATR1    00181*00194
710E STATR2    00186 00191*
719F STATT     00176 00206 00267*
71A2 STATT1    00268*00272
0026 STFLG     00042*00164 00172 00189 00193 00210 00275 00287
7099 STOP      00061 00083 00130*
0021 STOPB     00037*
7221 SUBTR     00119 00151 00348*
0019 T         00029*
730A TELP1     00598*00602
0010 TEM       00033*00557 00565
7052 TERM0     00095 00097*
7059 TERM1     00098 00100*
7062 TERM2     00101 00104*
7065 TERM3     00092 00103 00105*
730E TEST      00086 00583*
73B1 TEST1     00584*00592
73B4 TEST2     00586*00593
73C3 TEST3     00591 00594*
0028 TFLAG     00044*00090 00204 00220 00254 00425
711F TMODE     00099 00205*
001F TOUT      00035*00110
7203 TYLP1     00457*00497 00500
720A TYPE      00408 00416 00420 00424 00438 00452*
000C VAL       00018*00310 00318 00343
7178 VECT      00241 00246*
717B VECT1     00247*00253
7197 VECT2     00255 00262*00265
7199 VECT3     00248 00263*
73C7 VNO       00205 00597*
7000 VTBL      00058*00246 00252

```

APPENDIX D
FORTRAN-CALLABLE DRIVER AND FORTRAN TEST PROGRAM

```

      .LIST      TTM
*****
$ DLV11,MAC
$
$      DLV11 FORTRAN-CALLABLE DRIVER
$
$      CALL INITDL(ENDRD,ENDWR,BUFFER,BUFSIZ)          INITIALIZES
$
$      ENDRD: LOGICAL*1  END OF INPUT LINE CHARACTER
$      ENDWR: LOGICAL*1  END OF OUTPUT LINE CHARACTER
$      BUFFER: LOGICAL*1  BUFFER (ARRAY) FOR USE BY THIS HANDLER
$      BUFSIZ: INTEGER*2  BUFFER SIZE
$                      BUFSIZ >= M+1 WHERE M IS THE # OF CHARACTERS IN
$                      THE LONGEST MESSAGE (INCLUDING ENDWR)
$                      (BUFFER SHOULD BE LONG ENOUGH TO HANDLE ANY
$                      STACKING OF MESSAGES WHICH MAY OCCUR)
$
$
$      CALL CLRDL          CLEARS BUFFER OF ANY OLD INPUT
$
$      N = READDL(INLINE,IWAIT)          READS INPUT LINE
$                      (DECLARE READDL INTEGER*2 ! )
$
$      INLINE: LOGICAL*1 ARRAY WHERE CHARACTERS ARE STORED
$              (NOT INCLUDING ENDRD CHARACTER)
$      IWAIT: INTEGER*2  WAIT INDICATOR
$              0 - DO NOT WAIT FOR INPUT
$              1 - WAIT FOR INPUT (INDEFINITELY)
$      N      : INTEGER*2  NUMBER OF CHARACTERS READ IN,
$              NOT INCLUDING ENDRD
$              0 - NO INPUT HAS OCCURRED
$
$      CALL WRITDL(OUTLIN)          WRITES OUTPUT LINE
$
$      OUTLIN: LOGICAL*1  ARRAY CONTAINING CHARACTERS
$              TO BE SENT (ENDWR CHARACTER AT END IS NOT SENT)
$
$      NOTE: WRITDL HAS A POTENTIAL PROBLEM IN THAT OUTPUT IS
$            NOT BUFFERED, BUT IS UNDER INTERRUPT CONTROL
$            AFTER THE FIRST CHARACTER.  HENCE THE USER COULD
$            CHANGE THE CONTENTS OF AN OUTPUT LINE WHILE IT IS
$            BEING OUTPUT.  TO CIRCUMVENT THIS, DO NOT USE
$            THE SAME OUTLIN IN TWO CONSECUTIVE WRITDL CALLS.
$
$
$      SSJ 09-FEB-81
$      REV. SSJ 04-NOV-81  0935
$
*****

```

```

;
; .TITLE DLV11
; .GLOBL INITDL,CLRDL,READDL,WRITDL
;
; REGISTER DEFINITIONS AND SPECIAL ADDRESSES
;
;
; RVEC=310 ;RECEIVE VECTOR
; XVEC=314 ;TRANSMIT VECTOR
; RCSR=176510 ;RECEIVE CONTROL/STATUS REGISTER
;
; RVEC2=RVEC+2 ;RECEIVE PSW
; XVEC2=XVEC+2 ;TRANSMIT PSW
; RBUF=RCSR+2 ;RECEIVE BUFFER
; XCSR=RCSR+4 ;TRANSMIT CONTROL/STATUS
; XBUF=RCSR+6 ;TRANSMIT BUFFER
;
; PARAM=R5 ;PARAMETER REGISTER
; SAVPTR=R4 ;SAVE POINTER REGISTER
; COUNT=R0 ;COUNT OF INPUT CHARACTERS
; WAIT=R1 ;WAIT FLAG
;
;
; INITIALIZATION ROUTINE - SET UP INTERRUPTS,
; GET PARAMETERS
;
INITDL: CLR @#RCSR ;CLEAR RECEIVE STATUS
CLR @#XCSR ;CLEAR TRANSMIT STATUS
MOV #RINTER,@#RVEC ;SET UP RECEIVE INTERRUPT VECTOR
MOV #340,@#RVEC2 ;DISABLE ALL OTHER INTERRUPTS
MOV #XINTER,@#XVEC ;SET UP TRANSMIT INTERRUPT VECTOR
MOV #340,@#XVEC2 ;DISABLE ALL OTHER INTERRUPTS
MOVB @2(PARAM),REND ;GET END-OF-INPUT CHARACTER
MOVB @4(PARAM),XEND ;GET END-OF-OUTPUT CHARACTER
MOV @6(PARAM),BUFFER ;GET ADDRESS OF BUFFER
MOV @10(PARAM),BUFEND ;GET BUFFER SIZE
ADD BUFFER,BUFEND ;GET LAST ADDRESS OF BUFFER (+1)
JSR PC,CLRDL ;SET UP POINTERS
MOV #100,@#RCSR ;ENABLE THE RECEIVE INTERRUPT NOW
RTS PC ;RETURN
;
;
; CLEAR BUFFER ROUTINE
;
CLRDL: MOV BUFFER,RCPTR ;SET RECEIVE POINTER TO BEGIN OF BUFFER
MOV BUFFER,RDPTR ;SAME WITH READ POINTER
MOV BUFFER,LNPTR ;SAME WITH LINE POINTER
RTS PC ;RETURN

```

```

;
;
; READ ROUTINE
;
READDL: CLR      COUNT      ;RESET COUNT OF INPUT CHARACTERS
        MOV      2(PARAM),INPTR ;GET ADDRESS FOR FIRST CHARACTER
        MOV      @4(PARAM),WAIT ;GET WAIT FLAG
9$:      CMP      LNPTR,RDPTR  ;HAVE WE READ THE LATEST LINE ALREADY?
        BNE      3$          ;NO - A NEW LINE HAS COME IN
        TST      WAIT        ;YES - SHOULD WE WAIT?
        BEQ      7$          ;NO - USER WANTS AN IMMEDIATE RETURN
        BR       9$          ;YES - WAIT FOR A NEW LINE
3$:      INC      RDPTR        ;BUMP POINTER TO NEXT CHARACTER
        CMP      RDPTR,BUFEND ;ARE WE AT THE END OF THE BUFFER?
        BNE      1$          ;NO
        MOV      BUFFER,RDPTR ;YES - START AT BEGINNING OF BUFFER
1$:      CMPB     @RDPTR,REND   ;IS THIS THE END OF MESSAGE?
        BEQ      7$          ;YES - DONE, DON'T INCLUDE THIS CHAR
        MOVB     @RDPTR,@INPTR ;NO - MOVE THE CHARACTER TO USER'S LINE
        INC      COUNT        ;ADD 1 TO CHARACTER COUNT
        INC      INPTR        ;BUMP POINTER TO USER'S LINE
        BR       3$          ;REPEAT
7$:      RTS      PC          ;RETURN
;
;
; RECEIVE INTERRUPT ROUTINE
; OCCURS WHEN A CHARACTER HAS COME IN AND IS READY TO BE READ
;
RINTER: INC      RCPTR        ;BUMP RECEIVE POINTER
        CMP      RCPTR,BUFEND ;HAVE WE REACHED THE END OF BUFFER?
        BNE      1$          ;NO
        MOV      BUFFER,RCPTR ;YES - START AT BEGINNING AGAIN
1$:      MOVB     @RBUF,@RCPTR ;PUT CHARACTER IN BUFFER
        CMPB     @RCPTR,REND   ;IS THIS THE LAST CHAR IN LINE?
        BNE      3$          ;NO
        MOV      RCPTR,LNPTR  ;YES - MARK THE END OF THE LINE
3$:      RTI                ;RETURN
;
;
; OUTPUT ROUTINE
; USES TRANSMIT READY INTERRUPT FEATURE
; NOTE: EXPERIENCE WITH DLV11J SHOWS THAT THE READY INTERRUPT
; OCCURS IMMEDIATELY AFTER THE ENABLE
;
WRITDL: BIT      @*XCSR,#100   ;IS PREVIOUS OUTPUT FINISHED?
; (IE, IS INTERRUPT DISABLED?)
        BNE      WRITDL       ;NO - JUST WAIT HERE UNTIL IT IS
        MOV      2(PARAM),SAVPTR ;GET ADDRESS OF FIRST CHARACTER
        MOV      SAVPTR,TRPTR  ;SAVE FIRST ADDRESS FOR USE LATER
        MOV      #100,@*XCSR   ;ENABLE TRANSMIT READY INTERRUPTS
        NOP                     ;WAIT A BIT
        NOP
        CMP      TRPTR,SAVPTR  ;HAS READY INTERRUPT OCCURRED JUST NOW?
        BNE      5$          ;YES - OUTPUT HAS STARTED ALREADY
        JSR      PC,SEND      ;NO - SEND OFF FIRST CHARACTER
5$:      RTS      PC          ;RETURN
;

```



```

;
; TRANSMIT READY INTERRUPT ROUTINE
; OCCURS WHEN TRANSMITTER IS READY TO PUT OUT ANOTHER CHARACTER
;
XINTER: JSR      PC,SEND          ;SEND NEXT CHARACTER
        RTI              ;RETURN
;
;
; SEND A CHARACTER SUBROUTINE
;
SEND:   CMPB     @TRPTR,XEND      ;IS THIS THE END OF MESSAGE?
        BNE     7$              ;NO - SEND THIS CHARACTER
        CLR     @#XCSR          ;YES - DISABLE INTERRUPT
        BR      5$              ;AND DON'T SEND CHAR, BUT JUMP PTR
7$:     MOVB     @TRPTR,@#XBUF    ;MOVE CHARACTER OUT
5$:     INC      TRPTR           ;BUMP POINTER TO NEXT CHARACTER
        RTS      PC             ;RETURN
;
;
; CONSTANTS AND VARIABLES
;
;
REND:   .BYTE                      ;END-OF-INPUT CHARACTER
XEND:   .BYTE                      ;END-OF-OUTPUT CHARACTER
RCPTR:  .WORD                      ;RECEIVE POINTER (IN BUFFER)
RDPTR:  .WORD                      ;READ POINTER (IN BUFFER)
LNFPTR: .WORD                      ;END OF LAST LINE POINTER (IN BUFFER)
INPTR:  .WORD                      ;INPUT POINTER (IN USER'S INLINE)
TRPTR:  .WORD                      ;TRANSMIT POINTER (IN USER'S OUTLINE)
BUFFER: .WORD                      ;FIRST BUFFER ADDRESS
BUFEND: .WORD                      ;LAST BUFFER ADDRESS + 1
;
;
        .END

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C HLNCC
C
C     TEST INPUT AND OUTPUT TO HLNCC
C     FOR SWANSEN/ENSSLIN
C
C     JOHNSON 04-FEB-81
C     REV. 11-FEB-81 1130
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
0001      IMPLICIT INTEGER (A-Z)
C
0002      LOGICAL*1 CHAR(80),INLINE(80)
0003      LOGICAL*1 BUFFER (500)
0004      LOGICAL*1 LINEFD,BLANK
C
0005      DATA LINEFD/'12/,BLANK//  ''
C
C
C  USE THE NEW FORTRAN-CALLABLE DLV11 HANDLER
C
0006      5      CALL INITDL(LINEFD,BLANK,BUFFER,500)
0007      GO TO 20
C
C  LIST THE OPTIONS
C
0008      10      TYPE 1000
0009      1000 FORMAT(' ' - BEGIN COMMUNICATIONS//
X              ' S - START//
X              ' H - HALT//
X              ' Z - CLEAR SCALERS AND TIMER//
X              ' R - READ DATA TO COMPUTER//
X              ' P - READ OUT DATA TO HP97//
X              ' X - REQUEST STATUS//
X              ' I - SET INTERRUPT FLAG//
X              ' F - RETURN TO FRONT PANEL CONTROL///
X              ' T - TEST FOR INPUT//
X              ' W - WAIT FOR INPUT//
X              ' Q - REINITIALIZE INPUT//
X              ' O - LIST OPTIONS///)
C
C  GET OPERATOR INPUT
C
0010      20      TYPE 1001
0011      1001 FORMAT('/' $OPTIONS -> '/')
0012      ACCEPT 1002,CHAR
0013      1002 FORMAT(80A1)
C
C  TEST FOR SPECIAL CHARACTERS
C
0014      IF(CHAR(1) .EQ. 'O') GO TO 10
0016      IF(CHAR(1) .EQ. 'T') GO TO 50
0018      IF(CHAR(1) .EQ. 'W') GO TO 40

```

```

0020      IF(CHAR(1) .EQ. 'Q') GO TO 5
      C
      C  ASSUME CHARACTER ARE GOOD - SEND TO HLNCC
      C
0022      CALL WRITDL(CHAR)
0023      GO TO 20
      C
      C  READ IN RESPONSE FROM THE HLNCC
      C
0024      40  IWAIT = 1
0025      GO TO 55
      C
0026      50  IWAIT = 0
0027      55  N = READDL(INLINE,IWAIT)
      C
      C  CHECK THE RESPONSE
      C
0028      IF(N .EQ. 0) TYPE 1007
0030      1007 FORMAT('/' NO INPUT FROM HLNCC'/')
0031      IF(N .EQ. 0) GO TO 20
      C
      C  SEE IF THIS RESPONSE IS AN INTERRUPT FOR STATUS CHANGE
      C
0033      IF(N .EQ. 2) GO TO 100
      C
      C  THIS IS A DATA LINE - NOT AN INTERRUPT
      C
0035      TYPE 1008,(INLINE(1),I=1,N-1)
0036      1008 FORMAT('/' DATA FROM HLNCC:'//1X,80A1/')
0037      GO TO 20
      C
      C  THIS IS AN INTERRUPT
      C
0038      100 TYPE 1009,INLINE(1)
0039      1009 FORMAT('/' STATUS INTERRUPT FROM HLNCC:'//1X,A1/')
0040      GO TO 20
      C
0041      END

```

FORTRAN IV Storage Map for Program Unit ,MAIN.

Local Variables, .PSECT \$DATA, Size = 001240 (336, words)

Name	Type	Offset	Name	Type	Offset	Name	Type	Offset
BLANK	L*1	001225	I	I*2	001234	IWAIT	I*2	001230
LINEFD	L*1	001224	N	I*2	001232			

Local and COMMON Arrays:

Name	Type	Section	Offset	-----Size-----	Dimensions
BUFFER	L*1	\$DATA	000240	000764 (250.)	(500)
CHAR	L*1	\$DATA	000000	000120 (40.)	(80)
INLINE	L*1	\$DATA	000120	000120 (40.)	(80)

Subroutines, Functions, Statement and Processor-Defined Functions:

Name	Type	Name	Type	Name	Type	Name	Type	Name	Type
INITDL	I*2	READDL	I*2	WRITDL	I*2				