INTERFACING AN ARL PLASMA SPECTROMETER TO AN HP1000 MINICOMPUTER

By

R. C. BIGELOW

OPEN FILE REPORT 82-963

1982

INTERFACING AN ARL PLASMA SPECTROMETER TO AN HP1000 MINICOMPUTER

INTRODUCTION:

This paper describes a digital data link interface between an Applied Reasearch Labortories (ARL) 137000 Inductively Coupled Plasma emission spectrometer and a Hewlett Packard HP1000 minicomputer. The ARL is a self contained automated system that measures the concentration of up to 32 elements in liquid samples. A dedicated Digital Equipment Corp. (DEC) PDP11/04 minicomputer, or processor, controls the spectrometer, records the measured data on floppy disks, computes the element concentrations, and prints the results on an operator's terminal. A dialect of BASIC, written by the ARL staff, serves as both the programming language and operating system for the PDP controller. The HP1000 is a general purpose, multi-tasking, interrupt-driven, data acquisition system that is programmed in FORTRAN or Assembly Language and runs under RTE-IVB operating system with Multi-Terminal Monitor (MTM).

The digital data link eliminates manually keypunching and verifying the ARL data for entry into a data base on the USGS MULTICS system. Instead, data passes from the ARL over the data link to the HP1000 which reformats and stores it on 9-track magnetic tape for transport to MULTICS. The concept behind the data link is to have each minicomputer act as if the other is an intelligent terminal communicating at 1200 baud (120 characters per second) over standard, RS-232, hard-wired interfaces. This approach simplifies the hardware installation but creates some problems that must be solved by

software. First, coordinating programs must execute simultaniously in each processor to insure that the data is transmitted and received in proper sequence; second, at a more elementary level, each processor must control the rate at which it receives transmissions from the other. When a host computer communicates with non-intelligent terminals, it usually assumes that they are always listening for its output and that they can wait, if necessary, until the computer is ready to accept their input. Neither assumption is necessarily true when the "terminal" is intelligent or is another computer. To avoid information loss when two computers communicate, each must request permission of the other to transmit data and then wait for a go-ahead signal to insure that the recipient is ready to recieve more data. This process of controlled interchange, called "handshaking", solves the elementry timing problem; it is written into software for each processor. When the HP1000 wishes to transmit over the data link, it first sends ENQ (control-E) and, when the PDP11/04 is ready to accept input, it replies by sending an ACK (control F) character. The PDP requests to send with DC2 (control R) and the HP replies with DC1 (control Q) as an ok to send. Between handshakes, data is sent in blocks of characters whose maximum length is limited by the size of buffers in the receiving system. Buffers provide temporary storage for incoming data until it can be processed; buffer size is determined by hardware, software, or a combination of the two.

DATA LINK HARDWARE:

The data link hardware consists of RS-232-C interface cards in each processor, connecting plugs, and a three-wire cable between them (TABLE 1). Because the HP1000 is a multi-tasking, interrupt driven processor, its response time to an interrupt (request for service) from its interface cards varies unpredictably. The HP12966A interface card was chosen because it has a 128 character First-In-First-Out (FIFO) buffer on board to hold the input data. (Hewlett Packard,1979,1980). The buffer insures that data will not be lost while maintaining high throughput under usual working conditions. The FIFO accumulates data until it is half full, then it causes an interrupt to the HP Central Processing Unit (CPU) for service. The buffer continues to fill until either the CPU responds and empties it or an entire block of data is received. In the latter case, data transfer ceases until the CPU can respond. Usually, the CPU responds promptly and data from the link can be received almost continuously without loss. The same FIFO is used for both input and output which complicates its operation when fast turn around is required, as in handshaking (see below). The serial data rate (characters per second) can be set by software or by special wiring in the connector. The RS-232 byte format (number of bits and stop bits in a serial character) and the special (interrupting) characters that the 12966A will recognize are set by software commands. The commands set bits in on-board random access memories (RAM) whose outputs control the hardware.

The PDP is equipped with a DL-11-W full-duplex RS-232 board that has

single-character hardware buffers on its input and output ports. (Digital Equipment, 1972,1976). The format and serial data rate are set by hardware switches on the board. When a new character arrives in the single character input buffer, it overwrites the previous character in the buffer. The PDP must intercept each incoming character, test it to determine whether it is part of a handshake sequence, store it in the PDP's memory, and be in position to intercept the next character before that character is overwritten. This constraint gives rise to time-critical code in the PDP software.

DESCRIPTION OF DATA LINK SOFTWARE:

The software part of the data link consists of three user programs ("HP","ICP", and "FSEND") and a driver for the 12966A board ("DVX05"). The driver is specialized system software that controls the 12966A board; it takes standard system input, output, and control calls and implements them as well as handling input/output (I/O) protocols like the handshaking routine. It also defines the special characters and break characters for the board. Program "HP" runs on the PDP and enables its terminal (DEC terminal) on the ARL system to emulate a user terminal on the HP1000. With "HP" executing, the ARL operator can run the "ICP" program on the HP1000. Programs "ICP" and "FSEND" coordinate to transfer data from the the ARL to the HP1000. Program "ICP" signals the PDP to load and run "FSEND" and then waits for input. Program FSEND" retrieves emission data from a file stored on a floppy disk, calculates concentrations in parts per million and transmits the results over the data link along with header information that identifies the data. The

"ICP" program receives the headers and data, reformats them into 128 character blocks, and stores the blocks on a 9-track magnetic tape. When the process completes, the "HP" program is reloaded in the PDP. The forgoing sequence repeats for each file the ARL operator wants to transmit.

The software emphasises the differences between the two processors and their operating systems. In the HP, handshaking is done by a driver while data processing and inter-processor coordination are done by the "ICP" user program. It was not feasible for the author to write a driver for the ARL system operating under BASIC. Instead, handshaking, processor coordination, and data processing are done in user programs written in BASIC. ARL BASIC executes very slowly compared to compiled code; commands take from one to approximately eighteen milliseconds to complete. The data link transfers 120 characters per second or one character every 8.33 milliseconds. Therefore, it was necessary to find a subset of BASIC commands with minimal execution times (TABLE 2) from which to write input routines. From these commands, tightly written, time-critical input routines were devised that execute in less than 8.1 milliseconds on the PDP11/04.

Driver DVX05 is a modified version of a driver called DVQ05 that was written by Allen N. Tibbetts of HP to interface non-HP terminals to the HP1000 (Tibbetts, 1980). The modifications include the ENQ-ACK and DC2-DC1 handshaking routines, which may be suppressed at the HP operators request, and the added ability to use a carriage return (control M) to generate a "break" on the data line. The present use of ENQ, ACK, and DC1 in DVX05 follows standard HP practice; however, this use of DC2 as a handshake character does

not. In accordance with HP standards for writing drivers, the DVX05 code is in two parts, an initiation section (IX05) and a continuation-completion section (CX05). Within the initiation section, the SHOTi routine normally executes only after boot-up, and the SETIO only when the select code of the call differs from the previous invocations of the driver. The CNxx definitions are for parameters that can be set by CN or EXEC(3,... calls. The handshaking characters, carriage return, and line feed (Control J) are defined by IX05 as special characters that interrupt immediately when received from the data link.

The system calls the continuation-completion section to answer an interrupt from the 12966A board. A memory register on the board records the cause of the interrupt. The driver reads its contents to determine whether to continue a previous data transfer, complete a time-out, detect an error condition, respond to receipt of a special character, or to service a "break" command from the input device. If the 12966A has been left in a "read" condition by previous calls, which is the normal condition, any special character will cause an immediate interrupt as soon as it is received. If the special character is a "carriage return" (control-M) and it is part of an input string, it is treated as an end-of-line (EOL) character. If it is not part of an input string or input is not expected by the driver, a carriage return will cause a branch to the "break" routine and the HP1000 operating system will respond with a system prompt to the data link. The regular "break" interrupt is a hardware function. Driver DVX05 will process it normally if the data link is connected to a device that can assert it.

Most of the driver documentation is in the body of the source code including the meanings of the EQT and EQT extension table entries, of the CN parameters, and the instructions for generating the driver into an HP system. One problem that has been controlled but not eliminated in the operation of the 12966A board concerns the FIFO and a hardware counter that keeps track of the number of characters in the FIFO. The hardware counter generates interrupts that signal the system when the FIFO is half full, full, or empty. Unfortunately, the handshaking operation involves a software trick that causes the counter to register one more count than is in the FIFO after each handshake. Unless the counter is cleared after each set of transfers, the miscount is cumulative and, after 128 handshakes, the counter signals an overflow error to the system and the driver hangs up. The only way to clear the FIFO counter is to do a master reset which clears most of the board including the FIFO. The driver code now includes provision for this, but with the consequence that the counter is off by one count and signals "half full" when there are only 63 characters in the FIFO and "full" when there are 127 whenever hand shaking is in use. Thus, fewer than 127 characters, including handshakes, can be sent at a time. If handshaking is not in use, the count is accurate and a full 128 characters can be accommodated by the FIFO.

Program "HP" is programmed in BASIC to run on the PDP11/04; it interchanges information between the DEC terminal and the HP1000 in a way that makes the operation of the PDP transparent to the user (ie: it appears to the terminal user that he is talking directly to the HP1000). The program establishes two arrays, A, and B, as a double buffer and generates the control

characters that are needed later. The heart of the program is on lines 40-70. Two of the three loops are time-critical; one loop alternately looks for input from the DEC terminal and from the HP1000; the other loop accepts one input byte from the HP1000, checks whether it is a handshake character and stores it in either the A or B array. The slowest execution time of either loop is between 7.9 and 8.1 milliseconds. While one buffer array is being filled by an input data stream from the HP, a TRANSMIT command prints to the DEC terminal from the other array by direct memory access. An ENQ-ACK sequence initiates the transfer and disables keyboard entry from the DEC terminal. An ENQ character terminates the input stream and causes a branch to lines 200 et seq. As soon as the TRANSMIT completes, a new one begins with the roles of the buffers reversed. The program outputs an ACK to signal it is ready for more input, and returns to the input loop. The double buffering assures almost continuous printing to the DEC terminal of the input data stream. A DC1 in the input stream enables the DEC terminal keyboard and turns on the echo to the terminal printer. Carriage return from the keyboard is always passed to the HP1000; it acts either as a normal keyboard input terminator, or as a "break" character. Program "HP" is terminated by a CAN (control-X) character in the input stream that is transmitted by the "ICP" program to load and run "FSEND".

User program, "ICP" is written in FORTRAN and runs on the HP1000. It assembles the header information and data it recieves from "FSEND" over the data link into 128 character blocks for storage onto a 9-track magnetic tape. The last four words (eight characters) of each block are a terminator flag

consisting of octal 0,0,0,-1. Each magnetic tape file begins with two file header blocks: the first contains an ASCII header of up to 120 characters; while the second contains the number of sample analyses in the file, the number of elements analyzed in each sample and the two letter abbreviations of the analyzed elements in the order that they appear in the sample data blocks. The headers and data of the analyzed samples appear in pairs of blocks following the file headers. The first sample block contains a 72 character ASCII header identifying the sample and the date/time of analysis, followed by the data for the first fourteen elements. The second block contains data for up to twenty more elements. The data entry for each element is made up two parts; its symbol, an ASCII 'equal', 'star', or 'less-than' sign, and a numerical value. An equal sign implies the data is valid. A star implies that the data is valid, but is less than two and one half times the detection limit. A less-than sign implies that the element was not detected, in which case, the numerical value is the detection limit, not experimental data. Values are in parts per million; if a value of one million appears, it implies that the detector saturated and the concentration is too high to measure. The data is input under free field formatting which requires that the symbols be input in their octal equivalents preceeded by an '@' sign. Numerical values can range up to 9 digits and decimal point. Symbols are output to the magnetic tape in the form of an ASCII literals that are followed by numerical values in "E" format.

Program "FSEND" is written in BASIC and executes in the PDP; it cooperates with and supplies the data to program "ICP". The program is a

modified version of one written for the USGS by J. Hinthorne (unpublished program, 1981); both versions utilize subroutines from ARL supplied programs. Program "FSEND" retrieves the minimum detectable limits from subroutine 2000 and the list of elements detected by the ARL 137000 from a disk file. It asks the ARL operator to enter the name of a disk data file and opens that file. The operator is prompted for the names of elements to be omitted and for the name of the person requesting the analysis. With this information the program creates and sends the file header information to "ICP" and to the operator's terminal. The disk data file contains a series of records. Each record contains the results of measurements on a single sample preceeded by a header that identies the sample. For each record, the program first retrieves the header, sends it to the HP1000, and prints it on the operators terminal. Then, the program retrieves the data, compares it to the detection limits (lines 500-560), formats it, and sends it to the HP1000. Each data value, preceeded by the octal equivalent of its symbol is sent to the HP in 14 character blocks: (b@ssbn.nnnnnnn) where 'b' is blank, '@' is the octal sign, required by the HP format, 'ss' is the octal equivalent of the symbol and 'n' is the numerical data. The position of the decimal point changes, where necessary to match that of the data. Lines 3010-3040 contain the time-critical code for handshaking with the HP1000. Entry at line 3000 handshakes with DC2-DC1, entry at 2900 substitutes carriage return for DC2. When the program is finished, it loads program "HP" in response to an ENQ from the HP1000 and prints the messages on lines 3100 and 3105.

COMMENTS AND CONCLUSIONS:

The data link has proven to be a reliable, moderate speed method of transmitting data from the ARL induction coupled plasma emission spectrometer to the HP1000 minicomputer. The overall data rate is approximately 60 characters per second limited principally by the execution rate of the BASIC interpreter in the PDP11/04. At that rate, the results of a day's analyses can be transmitted to the HP1000 in 15 to 20 minutes which is sufficient for the needs of present users. The "ICP" program will be changed to create disk files in the HP1000 rather than go direct to tape as is now done. If a number of files representing several thousand data points are transmitted in one session, the magnetic tape unit is monopolized for several hours which is too long for the present system.

The following information is needed to operate the data link system. The ARL operating system imposes a limit of 80 characters per line on the output from port 3, which connects to the data link. The restriction must be eliminated by zeroing memory location 422 in the BASIC file on disk. A PATCH routine that is supplied by ARL will modify the file. The needed command sequence is almost correctly given in the BASIC manual (Griffith and others, 1978), as part of an example except that zero, not 200 must be put in location 422. BASIC must be bootstrap loaded from a modified disk for "HP" or "FSEND" to work properly, otherwise, the handshaking synchronization is destroyed by unexpected carriage returns every 80 characters.

The HP system manager must initialize DVX05 with a CN,lu,20B command,

where lu is the logical unit associated with the data link, just as if it were an interactive terminal. However, the PDP does not have to be on line at the time. Other CN commands of use are: CN,lu,30B,nn where nn is an octal number that determines the format of the serial interface; CN,lu,36B,tt where tt is a time-out value; and CN,lu,25B,h where h equals 1,2, or 3 to enable ENT-ACK, DC2-DC1, or both handshakes; if h is zero, there is no handshaking. Default values for these CN commands are not valid for the data link.

The following comments are intended for those who wish to adapt this software to other ARL-HP systems. Inevitably, hindsight reveals improvements that could be made in the design, and particularly in the programming of the data link. In accordance with Murphy's Law, the author did not learn of the limited execution rate of the BASIC interpreter until nearly the end of the design, nor did he learn that ARL has developed routines to exchange data with IBM computers until after the time-critical code was developed. Those routines support a different handshaking protocol from that used here; they may be found in an ARL supplied software library called EXGEN. The data exchange would be faster if the headers and data were sent unmodified to the HP1000 and the processing were done by it rather than by the PDP. The "HP" and "FSEND" programs were developed separately but might be combined, particularly if "FSEND" were reduced to minimum size. The feasibility of doing so depends on solving the problems associated with time-dependent code.

The CALL(8,Q) command in "HP" and "FSEND" is a function that fills the Q variable with octal characters. This function is no longer supplied with ARL BASIC; however, there is a similar call in EXGEN library.

Current model ARL Plasma units are controlled by PDP11/03 processors whose execution speeds for their current BASIC are unknown to the author. To use the present BASIC programs with another controller, a new set of timings for the time-critical commands must be found. A short program to determine execution times is given in Figure 1.

ACKNOWLEDGMENTS

Driver DVX05 is a modified version of a driver called DVQ05 that Allen N. Tibbetts of HP Neely Region wrote to support non-HP terminals on HP1000 systems. DVQ05 is not supported by the HP organization, but is available from some system engineers; the author acknowledges and thanks Shulom Kurtz and Dan Fishman of HP for supplying copies of it. Mr. Kurtz also made several valuable suggestions on developing driver systems and on system requirements. Having DVQ05 available saved the author months of development time.

Mr. James Hinthorne, formerly of ARL and now at Central Washington University, wrote a program for the USGS under contract from which much of the material in program "FSEND" is taken. His program helped the author to understand and use standard subroutines and functions that are part of the ARL induction coupled plasma system software. The ARL staff also very helpfully answered questions about their system and provided information on ARL system library calls.

Stanley Church provided the impetus for this interface project; Jerry Motooka, Lamont Wilch and Wendy Speckman patiently endured the inconveniences of testing the system.

REFERENCES

1. Digital Equipment Corp, 1972, DL11 asynchronous line interface manual, DEC-11-HDLAA-B-D: 4th Printing, Jan. 1975: Maynard Mass. Cable connections for the DL-11 series appearin chapter 3; The DL-11-W is not explicitly described but has the same connections.

2. ---, 1976, PDP11/04 System Users Manual, part no. EK-1104-OP-001:
Pages 4-19 show switch settings and vector addresses for the DL-11-W

3. L. Griffith, W. Hamilton, J. Hinthorne, S. Hooper, 1978, ARL EXTENDED BASIC language manual, Revision -100, April 5, 1979, Part No. 123498-100: Applied Research Labratories, Los Angeles. Contains utility programs such as PATCH as well as a very extended, non-standard form of BASIC.

4. Hewlett Packard Company, 1979, HP 12966A Buffered Asynchronous Data Communications Interface; Installation, Service, and Reference Manual, Part No. 12966-90001, Jan. 1979 Ed.: Data Systems Division, Cupertino, Calif.
Earlier editions omit the wiring for pin N in table 4-1; it should go to positive 5 volts.

5. ---, 1980, RTE Drivers DVR05/DVA05 For HP 263X/264X Terminals, Part no. 92001-90015, updated 1 October 1980.
The updated version explains how the standard drivers for the 12966A handle ENQ-ACK handshaking protocol

6. Allen N. Tibbetts, 1980, DVQ05-- Non protocol driver for 12966A 16 June 80: c/o Hewlett Packard Corp, 11000 Wolfe Rd., Cupertino Calif. 95014
Tibbetts has released a newer version of DVQ05, dated 25 Sept 80, that has his version of ENQ-ACK handshaking plus DC1 pacing for reads. The newer version will drive both HP and Non-HP terminals. It has no equivalent of DC2 since no HP terminals use that type of handshaking. Some of the warnings included in the DVX05 source code listed in the appendix have been taken from the newer version.

SOURCE CODE FOR TIMING ROUTINE IN BASIC

```
1 REM* TIMING ROUTINE--CODE TO BE TESTED GOES BETWEEN LINES 20 AND 7090.
2 REM* ALL BRANCHES IN CODE UNDER TEST MUST GO TO 7090.  X1 = EXECUTION
3 REM* TIME OF THE ROUTINE WITH NO TEST CODE. NORMALLY, TEST CODE EXECUTES
4 REM* 1000 TIMES BUT WILL LIMIT AT 30 SECONDS. IF M < 1000, DIVIDE GIVEN
5 REM* DIVIDE GIVEN M VALUE INTO 30 SECONDS FOR APPROX. CODE EXECUTION TIME.
6 REM* INITIAL CONSTANTS FOR TEST CODE GO HERE.
.
.
19 GOTO8000: REM* TIMING BEGINS @ '8010 CLOCK 0'. LINE 20 MUST STAY IN CODE
20 REM
.
.
.
.
7090GOTO8020
8000 LET M=-1:INTERRUPT 8040,30
8010 CLOCK 0
8020 LET M=M+1:IF M<1000 GOTO20
8030 CLOCK X:GOSUB8040:STOP CLOCK STOPPED
8040 CLOCK Y:USE 8.3:SEND(0)
8050 LET X1=7.55:REM* ZERO OFFSET FOR OVERHEAD
8060 PRINT"CODE EXECUTION TIME IN MILLISECONDS =" X/60-X1,"M = ";M
8070 RETURN
```

TABLE 1: INTERCONNECTION AND JUMPER LIST
FOR DL-11-W AND 12966A

BERG 40 PIN CONNECTOR ON DL-11-W			HP HOOD PLUG CONNECTOR ON 12966A	
SIGNAL NAME	PIN(S)		PIN(S)	SIGNAL NAME
signal ground	A,B,UU,VV	<------>	A,BB,1,24	signal ground
transmit data	F	)------>	S	receive data
recieve data	J	<------(	D	transmit data
jumpered pins	E,N		F,X,Y,AA	jumpered pins
jumpered pins	T,V		H,J	jumpered pins
jumpered pins	BB,DD		W,5	jumpered pins
			4,21	jumpered pins
			1,22	jumpered pins

The following pins on the 12966A hood plug must be wired as shown to to determine the data rate of the interface. For other speeds, consult the HP12966A interface reference manual.

SPEED	TIE TO PIN 8 (+5 V.)	TIE TO GROUND PIN(S)
1200 baud	Pins N,14,15	Pins 12,13
300 baud	Pins N,12,13	Pins 14,15
programmed in driver	(Pins 12-15 left open)	Pin N

The following switch settings must be made on the DL-11-W board to determine the speed and formatting of the serial data. For other speeds, consult the DL-11 manual

	1 2 3 4 5 6 7 8 9 10		1 2 3 4 5 6 7 8 9 10	
Sw1	1 1 0 1 0 0 1 1 0 1	Sw1	1 1 0 1 0 0 1 1 0 1	1 => on
Sw2	x x 0 1 1 0 1 0 ----	Sw2	x x 0 1 1 0 1 0 ----	
Sw3	1 0 0 0 1 1 0 1 0 1	Sw3	0 0 1 0 1 1 0 1 0 1	0 => off
Sw4	0 0 0 0 1 0 0 x x 1	Sw4	0 0 0 0 1 0 0 x x 1	
Sw5	0 0 1 0 1 1 0 1 1 0	Sw5	0 0 1 0 1 1 0 1 1 0	x => dont care
	FOR 1200 BAUD		FOR 300 BAUD	

TABLE 1: EXECUTION TIMES OF ARL BASIC COMMANDS

COMMAND	CONDITIONS	TIME milliseconds
10REM	no text	.28
20REM text	33 char. message	.55
30KEYJ,0	inputs a char. to J from port 0	1.72
40IF@D=""THEN40	@D not = to blank, branch not taken	1.23
(same)	@D = blank, branch taken	1.92
50IF@D()@STHEN7090	@D=* @S= space	2.15
(same)	@D=@S; @S= space	3.30
60KEYD,0:IF@D=""THEN40	branch not taken	2.77
(same)	branch taken	3.48
70SEND(7):PRINT@D;	stores @D in a buffer designated by port 7	3.47
70SEND(7):PRINT@D	note: no semicolon	17.45
80KEYJ,3:IF@H=""THEN7090 81SEND(5):PRINT@H;:IF@H=@DGOTO7090 82GOTO7090	\ worst) case, last / branch taken	8.05

Times are approximate and may vary by .2 milliseconds; where measurements varied, worst cases times are given.

APPENDIX

A. PROGRAM HP
B. PROGRAM ICP
C. PROGRAM FSEND
D. DRIVER DVX05

A. PROGRAM HP

```
PAGE  01    10:55:59 15-MAY-81

1REM DOES NOT USE DC2-DC1 PROTOCOL OR ECHO FROM HP
3 REM  VERSION 12
10DIMA(22),B(22)
11DIMQ(128)
12CALL(8,Q)
13LET E5=Q(5):LET A6=Q(6):LET D1=Q(17):LET D2=Q(18):LET S0=Q(32)
14LET Z=Q(0):ZERO A,B
15LET F=-1:BUFFER A,9
16LET @R=@Q(13)
17LET @L=@Q(10)
18SEND(9)
20LET @X=@Q(24)
30KEYD,0:IF@D<>""GOTO30
31LET @E=@E5
34REM
35REM
40KEYD,0:IF@D=@ZTHEN50
43SEND(0):PRINT@D;
44IF@D=@RTHENPRINT@L;
45SEND(3):PRINT@D;:SEND(9)
50KEY@H,3:IF@H=@ZTHEN40
52IF@H=@XGOTO1000
55PRINT@H;
60IF@H=@EGOTO200
65KEYH,3:IF@H=@ZTHEN40
70GOTO55
75REM
100REM
200REM
220LET F=-F
225IF F>0 THEN BUFFER B,9:TRANSMIT A,0
227IF F<0 THEN BUFFER A,9:TRANSMIT B,0
235IF@H=@E5THENLET@D=@A6:GOTO45
240GOTO50
1000SEND(3)
1010KEYH,3:IF@H=@ZTHEN1010
1015IF@H=@ETHENPRINT@A6;:GOTO1010
1020IF@H=@D1THEN1030
1025GOTO1010
1030SEND(0): REM RETURN TO KEYBOARD BEFORE EXIT
1040 RUN FSEND:  REM "DC1 SAYS HP IS WAITING FOR INPUT"
1050STOP ERR 1050

READY
```

B. PROGRAM ICP

```
FTN4,L
      PROGRAM ICP
      INTEGER H(60),E(4),Bi(36),EMENTS,SAMPS
      INTEGER CONTX,EL(56),V(6),FLAG(34),IOBUF(255),TERM(4)
      REAL DATA(34)
      DATA CONTX/30B/
      DATA DATA/  34*0.    /
      DATA EL  /28*020040B/
      DATA FLAG/ 34*000040B/
      DATA TERM/3*0B,177777B/
      CALL RMPAR(V)
      LU=V(1)
      CALL LGBUF(IOBUF,255)
      MT=8

      WRITE(LU,1000) CONTX
1000  FORMAT(R1)
      READ(LU,1001) H
      WRITE(MT,1001) H,TERM
1001  FORMAT(60A2,4I2)
2001  FORMAT(8(O6,1X)/)
      READ(LU,1001) EL
      READ(LU,*) EMENTS,SAMPS
      CALL CODE
      WRITE(E,1004) SAMPS,EMENTS
1004  FORMAT(2I4)
      WRITE(MT,1005) E,EL,TERM
1005  FORMAT(4A2,56A2,4I2)
      DO 20 I=1,SAMPS
      DO 10 J=50,72
10    Bi(J)=2H
      READ(LU,1016) Bi
1016  FORMAT(36A2)
      READ(LU, *) (FLAG(K),DATA(K),K=1,EMENTS)
      WRITE(MT,1009) Bi,(FLAG(K),DATA(K),K=1,4),TERM
      DO 30 L=4,24,10
      WRITE(MT,1010) (FLAG(K),DATA(K),K=L+1,L+10),TERM
30    CONTINUE
1009  FORMAT(36A2,4(R1,E11.5),4I2)
1010  FORMAT(10(R1,E11.5),4I2)
      CALL EXEC(3,LU)
20    CONTINUE
      ENDFILE MT
      END
$
```

C. PROGRAM FSEND

```
1REM FSEND=RSEND MODIFIED BY R.C.BIGELOW APR1981
2REM BY J. HINTHORNE  15-AUG-80
30PEN#SYMB,0:FETCH(1,0)T9
4 DIM G(12),G1(4),G2(2),G3(2)
5 DIM X(63),A(63),H(12),D(5),E(63),D1(63)
6DIME1(T9)
7 GOSUB 2000
11REM GET ELEMENT LIST FOR INSTRUMENT
12 AFETCH (2,0) (T9)E(1)
13 REM
20 LET J2=0
200PRINT:PRINT"SEND DATA FILE TO H.P."
210PRINT:PRINT"INPUT ANALYTICAL DATA FILE NAME ";:INPUT@H4
2110PEN@H4,1,Z:IFZ<>0THENLETH5=1:GOTO220
2120PEN@H4,0,Z:IFZ<>0THENLETH5=0:GOTO220
213PRINT"NAMED FILE NOT ON DISKS. ":GOTO210
215IFH5<>0THENIFH5<>1THEN210:REM CHECK DRIVE #
2200PEN @H4,H5,H6  :IFH6 = 0 THEN PRINT"FILE NOT FOUND":GOTO210
225 FETCH (1,H5) R: LET N=(R-5)/4:REM  GET NUMBER OF ANALYSES
230 AFETCH (2,H5) (63)X(1):REM GETO/O LIST
232GOSUB 800
235 GOSUB 4000
240SEND (0) :PRINT "ENTER ANALYSES REQUESTORS NAME"
241 INPUT @G1
247 FOR J2=0 TO 3 STEP 3: SEND (J2)
248 GOSUB 3000:REM SET UP HANDSHAKE WITH HP1000
249 PRINT "ICP FILE = M"@H4" DATA FOR  "@G1;
250 GOSUB 3000
252USE4:PRINT " NO. OF SAMPLES = ";N;
253PRINT" DATE ";:DATE,:PRINT" TIME ";:TIME,
255GOSUB 2900
260 NEXT J2
262SEND(3)
264 LET X9=0
266 FOR T=1 TO T9
267 REM PRINT ELEMENTS BEING REPORTED AND COUNT THEM
268IF X(T) >0THEN USE 2:PRINT@E(X(T));:LET X9=X9+1
269 NEXT T
270 GOSUB2900
275 USE 4: PRINT X9; N;
280 GOSUB 2900
285 LET K=0
290 FOR I = 5 TO R-1 STEP 4
295LET K=K+1
300FETCH(I,H5)S1,(12)H,(6)D,D2,R1:REM GET HEADING, DIL FAC. AND RATIO
301 FOR J2=0 TO 3 STEP 3:SEND(J2)
302 USE 4:PRINT K; D2; R1;"    ";
303USE 12:PRINT @H;
304 PRINT" DATE = "D(0);D(1);D(2);"  TIME = ";D(3);D(4);D(5);
305 GOSUB 2900
306 NEXT J2
309 REM FETCH CONCENTRATIONS
310 AFETCH (I+1,H5) (63) A(1)
330 LET J1=0
340 FOR J=1 TO T9
350 IF X(J) = 0 THEN 400
```

```
355 LET J1=J1+1
360  GOSUB 500:REM  GET FORMAT, FLAG AND VALUE (F1,F2,V)
370 USE 2: PRINT" @";@F1;" ";
380 USE 9.7:PRINTV;" ";
385 IF J=T9 THEN 420
390 IF INT(J1/8)=J1/8 THEN GOSUB3000
400  NEXT J
420 GOSUB 2900
440NEXT I
480BACK
490 GOTO 3105
500 LET V=A(J): LET @F1=75
501 REM 75 IS OCTAL VALUE FOR "="
505 IF V<2.5*D1(X(J)) THEN LET @F1=52
506 REM 52 IS OCTAL FOR "*"
510 IFV<D1(X(J)) THEN LET V=D1(X(J)):LET @F1=74
511 REM 74 IS OCTAL FOR "<"
520 LET V=V*D2
550 IF @F1<>"74"THENLET V=V*R1
560 RETURN
800PRINT"ENTER ELEMENTS TO BE OMITTED--<CR> TO END LIST"
805 FOR I=1 TO T9
810 INPUT @A: IF @A=""THEN 850: REM CR ENDS INPUT
818 REM FIND POSITION OF ELEM. TO BE DELETED IN X(J) LIST
819 REM AND SET THAT X(J) = 0 SO WE CAN SKIP IT LATER
820 FOR J=1 TO T9: IF @A = @E(X(J)) THEN LET X(J)=0:GOTO840
830NEXT J:PRINT@A" NOT PROPER SYMBOL":GOTO810
840NEXT I:PRINT"!!! NO GOOD!!!  YOU DELETED ALL ELEMENTS":STOP
850RETURN
2000 ENTRY   SET LOWER LIMIT VALUES BY CHANNEL NUMBER
2010RESTORE2050
2020 READ (T9)D1(1)
2050DATA.0023,.0023,.015,.170,.050
2051DATA.008,.0013,.390,.183,.010
2052DATA.095,.010,.007,.245,.210
2053DATA.018,.052,.072,.007,.031
2054DATA.013,.047,.294,.150,.270
2055DATA.131,.0003,.031,.061,.012
2056DATA.133,.015
2057 DATA0,0,0,0,0
2058 DATA0,0,0,0,0
2059 DATA0,0,0,0,0
2060RETURN
2900 LET @R9=@Q(13):GOTO3005
3000REM  HANDSHAKE HANDLER FOR HP
3004LET @R9=@Q(18)
3005 IFJ2=0 THEN PRINT"":RETURN :  REM FOR DEC TERM
3007SEND(3):PRINT@R9;
3010KEYY,3:IF@Y=@ZGOTO3010
3020IF@Y=@C1THENRETURN
3030IF@Y=@E5GOTO3100
3040GOTO3010
3100 SEND(0):PRINT "HP WANTS TO TRANSMIT"
3105 PRINT"PROGRAM- HP - WILL LOAD AND RUN"
3110RUN HP
3120 STOP ERR3120
```

```
4000 DIM Q(64)  :REM FOR OCTAL CODES 000 TO 100B.
4005 CALL (8,Q) :REM PUT THE CODES IN
4010 DIM Q1(128):REM Q1 IS GENERAL CHAR. BUFFER
4020REM Q(5)  IS ENQ
4021REM Q(6)  IS ACK
4022REM Q(7)  IS BELL
4023REM Q(10) IS LF
4024REM Q(13) IS CR
4025REM Q(17) IS DC1
4026REM Q(58) IS :
4030LET C=Q(64) :  REM THIS IS THE @ SIGN NEEDED BY HPFOR OCTAL INPUT
4031LET C9=Q(32)
4032 LET E5=Q(5):  REM ENQ CHAR
4033 LET A6=Q(6) :  REM ACK CHAR
4034 LET Z=Q(0)
4035 LET C1=Q(17) :  REM DC1
4036 LET C2=Q(18) :  REM DC2
4050 RETURN

READY
```

D. DRIVER DVX05

```
0001                ASMB,R,L
0003  00000                 NAM DVX05,0 HP12966 DRIVER FOR 21MX REV 21, 04JAN82,R
0004                        ENT IX05,CX05
0005                        EXT $UPIO,$LIST
0006                       IFN          !!!!!DEBUG
0007                       EXT TRAP
0008                       UNL
0009                       XIF          !!!!!
0010*
```

```
0012%     COMMENTS BY R.C.BIGELOW 04 JAN 1982: VX5.21 VERSION
0013% THIS IS DRIVER DVX05.  IT  IS  DESIGNED  FOR  COMMUNICATIONS
0014% BETWEEN  AN  HP1000 WITH A 12966A I/O CARD AND  A  PDP-11/04
0015% CONTROLLER  RUNNING  BASIC  AS  PART  OF AN ARL  INDUCTIVELY
0016% COUPLED PLASMA  SPECTROMETER.  DVX05  MAY  BE  USABLE  WITH
0017% PROCESSORS  OR  CONTROLLERS OTHER THAN THE PDP IF  THEY  CAN
0018% SUPPORT THE TWO WAY HANDSHAKING BUILT INTO IT.  THIS VERSION
0019% OF DVX05 IS BASED  ON  ALAN  N TIBBETTS DRIVER CALLED DVQ05.
0020% DVX05 MAINTAINS ALL OF THE FEATURES THAT WERE IN DVQ05 AS OF
0021% JUNE  1980  AND  ADDS  THREE  OTHERS;  ENT-ACK  HANDSHAKING,
0022% DC2-DC1  HANDSHAKING,  AND  CARRIAGE  RETURN  AS  A  BREAK
0023% CHARACTER.  ENT-ACK IS SIMILAR TO  HP'S HANDSHAKING FOR 264X
0024% TERMINALS.  THE  12966A BOARD TRANSMITS AN "ENQ" CHARACTER AS
0025% A  REQUEST TO SEND AND WAITS FOR AN "ACK"  CHARACTER  BEFORE
0026% TRANSMITTING  ANY  CHARACTERS.  "ENQ" IS  TRANSMITTED  AFTER
0027% EVERY CARRIAGE RETURN AND LINE FEED (CRLF) OR AFTER EVERY 64
0028% CHARACTERS  WHICH EVER COMES FIRST.  DC2-DC1 HANDSHAKING  IS
0029% SIMILAR  EXCEPT THAT THE PDP11 SENDS  "DC2"  AND  WAITS  FOR
0030% "DC1" BEFORE TRANSMITTING.  THE ARL CONTROLLER HAS NO WAY TO
0031% GENERATE  A  "BREAK"  CHARACTER UNDER PROGRAM CONTROL TO GET
0032% THE ATTENTION  OF THE HP SYSTEM.  THIS DRIVER RESPONDS TO AN
0033% UNSOLICITED CARRIAGE RETURN, (CR), AS THOUGH IT WERE A BREAK
0034% CHARACTER  PROVIDED  THE  BOARD WAS LEFT IN A "RECEIVE" MODE
0035% AFTER  THE  PREVIOUS  OPERATION,  AS  IT  USUALLY  IS.  ON
0036% INTERRUPT, THE SPECIAL CHARACTER FLAG IS CHECKED BEFORE  THE
0037% BUFFER HALF FULL FLAG.  IF A "CR" INTERRUPTS THE DRIVER WHEN
0038% IT DOES NOT EXPECT INPUT, OR IF A "CR" IS RECEIVED  IN PLACE
0039% OF AN "ACK"  DURING  OUTPUT  HANDSHAKING,  THEN  THE  DRIVER
0040% BRANCHES TO THE BREAK ROUTINE.  OTHERWISE,  "CR"  IS A LINE
0041% TERMINATOR ON INPUT.  THERE WAS AN UNRESOLVED PROBLEM  WITH
0042% THE  ON-BOARD CHARACTER COUNTER THAT WAS CAUSED BY THE "FIFO
0043% TURN AROUND TRICK".  DETAILS ARE GIVEN IN  THE COMMENTS NEAR
0044% LINE 1412.  THE  CHARACTER  COUNTER  IS SUPPOSED TO INDICATE
0045% HOW MANY CHARACTERS ARE IN THE FIFO  AT  ANY  TIME;  HOWEVER,
0046% THE  COUNTER  IS ALWAYS ONE COUNT TOO HIGH AFTER INPUT  WITH
0047% HANDSHAKING.  PERFORMING A MASTER RESET  CLEARS  THE COUNTER
0048% BUT  THAT CANNOT BE DONE UNTIL THE BUFFER IS EMPTY SINCE  IT
0049% GETS CLEARED TOO!  THEREFORE, A  MASTER  RESET IS DONE AFTER
0050% THE ENQ-ACK HANDSHAKE AND JUST BEFORE THE "DLAY" PART OF THE
0051% "TRIGR" SUBROUTINE TO ELIMINATE THE SPURIOUS COUNT.  BECAUSE
0052% THE COUNTER CANNOT BE TRUSTED,  THE  INPUT ROUTINE TESTS FOR
0053% AN "INVALID CHARACTER" ON INPUT TO DETERMINE  WHEN  THE FIFO
0054% IS EMPTY.  SO FAR AS IS  NOW KNOWN, THE PROBLEM IS RESOLVED;
0055% HOWEVER, IF IT SEEMS TO  REAPPEAR, TRY "EXEC(3,LU)" CALLS IN
0056% THE USER PROGRAM TO  CLEAR  THE  DRIVER  PERIODICALLY.  THE
0057% FOLLOWING NOTES ARE  INCLUDED  FOR THOSE WHO WANT TO FURTHER
0058% MODIFY OR MAINTAIN  THIS  DRIVER.  I GIVE THE SAME "BIG BOY
0059% WARRANTY" THAT TIBBETTS DOES (SEE BELOW).
0060%
0061%        R.C.BIGELOW
0062%        U.S.GEOLOGICAL SURVEY M/S 918
0063%        BOX 25046, FEDERAL CENTER
0064%        DENVER COLORADO, 80225
0065%        (303) 234 2943   FTS SAME #
0066%
```

```
0068*   NOTES:
0069* "DC2", "ACK", AND "CR" ARE RECOGNIZED AS SPECIAL CHARACTERS
0070* THAT INTERRUPT.  "ACK" IS IGNORED IF IT IS NOT PART OF AN
0071* "ENQ-ACK" SEQUENCE.  "DC2" IS NOT PASSED TO A USER BUFFER
0072* AND WILL ACT AS AN INPUT SEQUENCE SEPARATOR. "CR" IS NOT
0073* PASSED TO A USER BUFFER AND ACTS AS AN INPUT SEQUENCE
0074* TERMINATOR. HANDSHAKING IS CONTROLLED BY BITS IN EQX13. IF
0075* BIT ZERO IS SET, DVX05 WILL ALWAYS ISSUE A "DC1" AFTER
0076* INITIALIZATION. IT REPLIES "DC1" TO "DC2" BUT IT DOES NOT
0077* DEMAND THAT "DC2" BE SENT BEFORE IT WILL ACCEPT CHARACTERS.
0078* IF BIT 1 IS SET THE DRIVER INSISTS ON "ACK" AS A REPLY TO
0079* "ENQ" BEFORE CONTINUING. BECAUSE OF THE HARDWARE LOCKOUT
0080* FEATURE ON THE INTERRUPTS ONLY THE FIRST UNANSWERED
0081* INTERRUPT IS ANSWERED BY THE SYSTEM, OTHERS ARE IGNORED.
0082* BINARY DATA EXCHANGE (AS OPPOSED TO ASCII) IS NOT REALLY
0083* ENVISIONED WITH DVX05. TO WRITE A DRIVER FOR BINARY DATA
0084* INTERCHANGE, IT MAY PAY TO CLEAR OUT THE SPECIAL CHARACTER
0085* RANDOM ACCESS MEMORY (RAM) FOR THE DURATION OF THE BINARY
0086* TRANSFER. "CR" AND "DC2" SHOULD NOT BE USED AS PART OF A
0087* BINARY TRANSFER SINCE THEY ARE NOT PASSED TO USER BUFFERS.
0088* AN INTERRUPT BY A "BREAK" WILL ALWAYS WORK TO GET OUT OF THE
0089* DRIVER, EVEN WHEN "CR" DOESN'T. THERE IS A PROVISION FOR
0090* BEING SYSTEM CONSOLE IN THE BREAK ROUTINE THAT HAS NOT BEEN
0091* USED OR INVESTIGATED. BREAK CAN SCHEDULE A PROGRAM BUT DOES
0092* NOT AT PRESENT DO SO. THE FACT THAT THE "SETIO" AND "SHOT1"
0093* SUBROUTINES ARE BYPASSED AFTER THE FIRST CALL TO THE DRIVER
0094* CAN BE A PROBLEM DURING DRIVER DEVELOPMENT, PARTICULARLY
0095* WHEN THE DRIVER TERMINATES ABNORMALLY. WHEN THE SYSTEM IS
0096* BOOTED UP, THE "SHOT1" ROUTINE GETS THE ADDRESS FOR PROGRAM
0097* "PRMPT" FROM SYSTEM MEMORY, SAVES IT IN "EQT01", AND
0098* REPLACES IT WITH THE ADDRESS OF THE EQT TABLE. IF "SHOT1"
0099* IS EXECUITED WITHOUT RE-BOOTING THE SYSTEM, THE ADDRESSES
0100* ARE MISPLACED. USEFUL HALTS FOR DEVELOPMENT WERE "HLT 55B"
0101* IN SHOT1, "HLT 22B" AND "HLT 20B" OF WHICH ONLY THE FIRST
0102* REMAINS ACTIVE. WITH THE EXCEPTION OF "CR", UNSOLICITED
0103* INTERRUPTS BY SPECIAL CHARACTERS ARE IGNORED. THIS KEEPS
0104* "DC2" INPUTS FROM INADVERTANTLY CALLING PROGRAM "PRMPT".
0105* PROGRAM "PRMPT" LINKS THE HP SYSTEM TO INTERACTIVE TERMINALS
0106* AND HAS ITS OWN QUIRKS. IT ONLY ALLOWS DRIVERS OF TYPE 00,
0107* 05, OR 07 TO BE INTERACTIVE AND FOR TYPE 05 DRIVERS THE
0108* SUBCHANNEL MUST BE ZERO OR IT WILL IGNORE THE CALL. THIS IS
0109* THE REASON HANDSHAKING IS ENABLED BY "EXEC 3" CALLS OR "CN"
0110* COMMANDS RATHER THAN BY SUBCHANNELS. UNLIKE DVQ05, THIS
0111* DRIVER WILL NOT WORK ON HP 2100'S. THE 2100 CODE WAS
0112* REMOVED TO ALLOW THE "IFN" OPTION FOR A DEBUG ROUTINE CALLED
0113* "TRAP" THAT PRINTED OUT THE A,B,S REGISTERS AND THE TRAP
0114* ADDRESS RELATIVE TO THE BEGINNING OF THE PROGRAM. I AM NOT
0115* PUBLISHING "TRAP" BECAUSE IT HAS AN INTERMITTANT ERROR.
0116* HP2100 CODE COULD BE RE-INSERTED IN THE DRIVER. IT PROBABLY
0117* SHOULD GO JUST WHERE THE 16 JUNE 1980 VERSION OF DVQ05 HAS
0118* IT.
```

```
0120* TO  GENERATE THE DRIVER INTO A SYSTEM RELOCATE  THE COMPILED
0121* CODE (EXAMPLE ENTRIES ASSUME A SELECT CODE OF 17)
0122*
0123*                REL,DVX05::32767   *  NON-HP TERMINAL DRIVER
0124*
0125*                17,DVX05,X=13      *  EQT TABLE ENTRY
0126*
0127* RELOCATE  EITHER MULTI-TERMINAL MONITOR  OR SESSION SOFTWARE
0128* AND SPECIFY:
0129*
0130*                 17,PRG,PRMPT          INTERRUPT TABLE ENTRY
0131*
0132* THE FOLLOWING COMMANDS  MUST BE GIVEN BY  THE SYSTEM MANAGER
0133* TO  INITIALIZE DVX05  AFTER  THE SYSTEM HAS BEEN  BOOTED-UP.
0134*
0135*         CN,LU,30B,040011B      1200 BAUD, NO PARITY, NO ECHO
0136*         CN,LU,25B,3B           ENT-ACK AND DC2-DC1 HANDSHAKE
0137*         CN,LU,20B              INITIALIZE  PDP AS A TERMINAL
0138*
0139* THE DRIVER HAS NOT  BEEN  TESTED  WITH SESSION MONITOR, ONLY
0140* WITH  MTM.  TIME-OUT HANDLING, MODEM  CONTROL  AND  PROGRAM
0141* SCHEDULING ON RECEIPT  OF  BREAK  HAVE  NOT BEEN MODIFIED OR
0142* TESTED EXCEPT AS NOTED ABOVE.
0143*
0144*       R.C.B.
0145*
0146*
0147*        THE FOLLOWING ARE TIBBETTS COMMENTS FOR DVQ05:
0148* THIS IS A 'PROTOCOL-LESS' DRIVER  FOR  THE 12966 CARD.  IT IS
0149* USED WITH SILENT 700'S, TTY'S, 'DUMB' TERMINALS, ETC.  IT CAN
0150* ALSO BE  USED  WITH  THE  HP7221  PLOTTER, WHICH REQUIRES DC1
0151* PACED  READS, BY USING SUBCHANNELS HAVING  BIT  3  SET.  THE
0152* INTENTION OF THIS DRIVER IS TO SERVE AS A STARTING POINT  FOR
0153* THOSE WHO WISH TO DEVELOP OTHER DRIVERS FOR THE 12966.  IT IS
0154* NAMED  DVQ05,  BECAUSE  THE  GRAPHICS ROUTINES FOR  THE  7221
0155* REQUIRE A '05' DRIVER AND Q IS ARBITRARY.  IT  IS  CLOSER  IN
0156* ITS ACTIONS TO DVR00 THAN DVR05.
0157*
0158*
0159*          THIS DRIVER CARRIES THE 'BIG BOY' WARRANTY;
0160* IF IT DOES NOT WORK  RIGHT,  YOU  ARE  A BIG BOY NOW AND WILL
0161* HAVE TO FIX IT YOURSELF.  HP WILL  NOT SUPPORT THIS DRIVER IN
0162* ANY WAY.  IF YOU NEED THE HELP OF A SYSTEMS ENGINEER, IT MUST
0163* BE  ON  A  TIME  AND MATERIALS BASIS.  ANY USE OF THIS DRIVER
0164* CONSTITUTES ACCEPTANCE OF THE FOREGOING CONDITIONS.
0165*
0166*     WRITTEN BY:
0167*     ALAN N. TIBBETTS
0168*     HP NEELY S.E.O.
0169*
```

```
0171                    IFN           !!!!!DEBUG
0172                    LST
0173                ORG DEF *
0174                    XIF           !!!!!
0175*
0176  00000        AREG  EQU 0
0177  00001        BREG  EQU 1
0178*
0179*  EQUIPMENT TABLE USAGE:  (CONFORMS TO DVR00 & DVA05 WHERE POSSIBLE)
0180*
0181  01660        EQT01 EQU 1660B     I/O REQUEST LIST POINTER
0182  01661        EQT02 EQU 1661B     INITIATION SECTION ADDRESS
0183  01662        EQT03 EQU 1662B     CONTINUATOR SECTION ADDRESS
0184  01663        EQT04 EQU 1663B     D BPS TUU UUU CCC CCC
0185  01664        EQT05 EQU 1664B     A AEE EEE ESS SSS SSS
0186  01665        EQT06 EQU 1665B     C CXX XBX EXX XXX XTT
0187  01666        EQT07 EQU 1666B     REQUEST BUFFER ADDRESS (WORD)
0188  01667        EQT08 EQU 1667B     REQUEST BUFFER LENGTH (WORDS OR BYTES)
0189  01670        EQT09 EQU 1670B     CURRENT BYTE ADDRESS IN BUFFER
0190  01671        EQT10 EQU 1671B     CURRENT REMAINING CHAR. COUNT (-)
0191  01672        EQT11 EQU 1672B     CO-ROUTINE ADDRESS
0192  01771        EQT12 EQU 1771B     EQT EXT SIZE !NOTE CHANGE HERE!
0193  01772        EQT13 EQU 1772B     EQT EXTENSION ADDRESS
0194  01773        EQT14 EQU 1773B     TIME-OUT RESET VALUE
0195  01774        EQT15 EQU 1774B     TIME-OUT CLOCK
0196*
0197  00000 000000 EQX01 NOP           BAUD SETTING & HALF DUPLEX FLAG
0198  00001 000000 EQX02 NOP           ID SEGMENT ADDR. OF PROGRAM TO SCHEDULE
0199  00002 000000 EQX03 NOP           LAST WORD OUTPUT TO CARD
0200  00003 000000 EQX04 NOP           PACE CHARACTER FOR READS
0201  00004 000000 EQX05 NOP           FORM FEED DELAY TIME (-CENTISECONDS)
0202  00005 000000 EQX06 NOP           CHARACTER LENGTH CONTROL WORD
0203  00006 000000 EQX07 NOP           CRLF DELAY TIME (-CENTISECONDS)
0204  00007 000000 EQX08 NOP           LAST LIA FROM CARD
0205  00010 000000 EQX09 NOP           REQUEST BUFFER ADDRESS (BYTE)
0206  00011 000000 EQX10 NOP           CURRENT LINE STATUS
0207  00012 000000 EQX11 NOP           REQUEST CODE FOR PACED WRITES
0208  00013 000000 EQX12 NOP           END OF BUFFER + 1
0209  00014 000000 EQX13 NOP           HANDSHAKING BITS REGISTER
0210*
0211  00015        XEQTL EQU *-EQX01    NUMBER OF WORDS IN EQT EXTENSION
0212                    IFN           !!!!!DEBUG
0213                    UNL
0214                    XIF           !!!!!
0215*  0 MEANS MUST BE ZERO, 1 MEANS MUST BE ONE, X MEANS DON'T CARE


0217********************** E Q T 0 4  BIT MAP **********************
0218*  D BPS TUU UUU CCC CCC
0219*    D       DMA USED
0220*    B       AUTO BUFFERING BY SYSTEM
0221*    P       DRIVER PROCESSES POWER FAIL
0222*    S       DRIVER PROCESSES TIME-OUTS
0223*    T       TIME-OUT HAS OCCURED (SYSTEM SETS ZERO BEFORE EACH CALL)
0224*    UU UUU  SUBCHANNEL NUMBER
```

```
0225*          ^-0-
0226*         ^--1-
0227*        ^---2-
0228*      ^------3- PACE READS WITH A TRIGGER CHARACTER (DC1 DEFAULT)
0229*     ^-------4-
0230*    CCC CCC I/O SLOT NUMBER
0231*************************** E Q T 0 5  BIT MAP ***************************
0232*  A AEE EEE ESS SSS SSS
0233*  A A       0=AVAILABLE 1=DISABLED 2=BUSY 3=WAITING DMA
0234*  EE EEE E  EQUIPMENT TYPE CODE
0235*  00 010 1  TYPE 5 TO FOOL GRAPHICS, ETC.
0236*  SS SSS SSS                                  CLEARED IN INITIATOR
0237*           ^--0-TIME-OUT IS EXPECTED                    *
0238*          ^---1-TERMINAL PROGRAM ENABLED (PRMPT)
0239*         ^----2-SPEED SENSING                          *
0240*       ^------3-PARITY ERROR ON LAST READ              *
0241*      ^-------4-DATA SET NOT READY                     *
0242*     ^--------5-CNTRL D ENTERED  (NOT IMPLEMENTED)     *
0243*   ^----------6-WAITING FOR PHONE TO RING             *
0244*  ^-----------7-BUFFER FLUSH (NOT IMPLEMENTED)
0245*************************** E Q T 0 6  BIT MAP ***************************
0246*  C CXX XBX EXX XXX XTT
0247*  C C       TYPE OF CALL--) 0=STD 1=BUFFERED 2=SYSTEM 3=CLASS
0248*  B         BINARY TRANSFER
0249*  E         ENABLE ECHO ON INPUT
0250*  TT        TYPE OF CALL 0=UNDEFINED 1=READ 2=WRITE 3=CONTROL
0251*     FOR BINARY READS, ALL 8 BITS OF ALL CHARACTERS EXCEPT THE
0252*     TERMINATOR BYTE (CR) WILL BE TRANSFERRED TO THE USER BUFFER.
0253*     FOR ASCII READS, THE CHARACTERS ARE MASKED TO 7 BITS, AND THE
0254*     CHARACTERS NULL, LINEFEED, RUBOUT, BACKSPACE AND THE TERMINATOR
0255*     BYTE WILL NOT APPEAR IN THE USER BUFFER.
0256*  0 MEANS MUST BE ZERO, 1 MEANS MUST BE ONE, X MEANS DON'T CARE


0258*************************** E Q X 0 1  BIT MAP ***************************
0259*  0 H00 XXX 000 00B BBB
0260*  B BBB = BAUD SETTING
0261*  H = HALF DUPLEX (NO ECHO) BIT
0262*************************** E Q X 0 6  BIT MAP ***************************
0263*     0 011 XXX XXX S0P PCC
0264*  S  0==) 1 STOP BIT   1==) 2 STOP BITS (1.5 @ 5 BITS/CHAR)
0265*  PP 0X ==) NO PARITY GENERATION OR CHECKING
0266*     10 ==) ODD PARITY GENERATION AND CHECKING
0267*     11 ==) EVEN PARITY GENERATION AND CHECKING
0268*  CC 00 ==) 5 DATA BITS IN CHAR.
0269*     01 ==) 6 DATA BITS IN CHAR.
0270*     10 ==) 7 DATA BITS IN CHAR.
0271*     11 ==) 8 DATA BITS IN CHAR.
0272*  NOTE THAT CC AND PP ARE INTERACTIVE.  SEE NOTE @ CN30
0273*
0274*************************** E Q X 1 3  BIT MAP ***************************
0275*
0276* X XXX XXX XXX XXX XED
0277*
0278*  IF BIT D IS SET DC2-DC1 HANDSHAKING IS ENABLED, OTHERWISE NOT
```

```
0279*  IF BIT E IS SET ENT-ACK HANDSHAKING IS ENABLED, OTHERWISE NOT
0280*
0281*****************************************************************************
0282*
0283                    IFN         !!!!!DEBUG
0284                    LST
0285                    XIF         !!!!!
```

```
0287  00015 000000  IX05  NOP           ENTRY TO INITIATOR
0288*     HLT 20B
0289                      IFN
0290                      JSB TRAP
0291                      DEF *-ORG
0292                      XIF
0293  00016 016374R       JSB SETIO     CONFIGURE I/O INSTRUCTIONS
0294*     HLT 21B       (( FOR DEBUG
0295                      IFN
0296                      JSB TRAP
0297                      DEF *-ORG
0298                      CLA           FOR DEBUG, FORCE THE USE
0299                      STA EQX02,I   OF THE SHOT1 ROUTINE
0300                      XIF
0301  00017 162001R       LDA EQX02,I   GET ID SEGMENT ADDRESS FROM EQT
0302  00020 002003        SZA,RSS       IS IT ALREADY SET-UP?
0303  00021 026105R       JMP SHOT1     NO, GO TO 1-SHOT CODE
0304  00022 102200  SFC1  SFC 00        IF THE FLAG IS SET ON THE CARD, THEN
0305  00023 026137R       JMP CLR       THIS IS THE 1ST ENTRY AFTER POWER-FAIL
0306*
0307  00024 017122R RFLGS JSB CSTAT     CLEAR STATUS BITS FOR TIME-OUT,PHONE,
0308  00025 177602        OCT 177602    SPEED SENSE,PARITY,DATA SET, & CNTRL D
0309  00026 016222R       JSB CNWD      ISOLATE REQUEST TYPE
0310  00027 000003  K3    OCT 000003    FROM LOW 2 BITS OF CONWD
0311  00030 026277R       JMP RJCT2     =0, WHICH IS UNDEFINED!
0312                      IFN
0313                      JSB TRAP
0314                      DEF *-ORG
0315                      XIF
0316  00031 052027R       CPA K3        IS IT A CONTROL REQUEST?
0317  00032 026231R       JMP CNTRL     YES
0318  00033 073275R       STA PRAM1     SAVE TYPE IN A TEMPORARY LOCATION
0319  00034 017171R       JSB STATS     READ LINE CONDITIONS
0320  00035 017156R       JSB LINCK     ARE THEY OK?
0321  00036 026077R       JMP IXDWN     NO, GO DOWN
0322*
0323* COMMON SETUP FOR READ OR WRITE CALLS
0324*
0325  00037 161666        LDA EQT07,I   GET REQUEST ADDRESS
0326  00040 000066        CLE,ELA       MAKE INTO BYTE ADDRESS
0327  00041 171670        STA EQT09,I   AND SAVE AS CURRENT POINTER
0328  00042 172010R       STA EQX09,I   AND ORIGINAL POINTER
0329  00043 161667        LDA EQT08,I   GET REQUEST LENGTH
0330  00044 002020        SSA           IF NEGATIVE, BYTE COUNT ALREADY
0331  00045 026050R       JMP STLEN     SO SAVE IT
0332  00046 001200        RAL           POSITIVE WORDS MUST BE MULT. BY 2
0333  00047 003004        CMA,INA       AND MADE NEGATIVE FOR # CHAR.
0334  00050 171671  STLEN STA EQT10,I   SAVE AS REMAINING COUNT
0335                      IFN
0336                      JSB TRAP
0337                      DEF *-ORG
0338                      XIF
0339*
0340  00051 016203R       JSB MCLR      RESET CARD AND PUT IN TRANSMIT MODE
0341* THIS ZEROS THE HARDWARE COUNTER ON THE CARD FOR READS OR WRITES
```

```
0342*
0343  00052 063275R       LDA PRAM1     FUNCTION =1 FOR READ, =2 FOR WRITE
0344  00053 052745R       CPA K1        FUNCTION CODE WAS
0345  00054 026061R       JMP READ      =1, A READ CALL




0347  00055         WRITE EQU *         =2, A WRITE CALL
0348  00055 062060R       LDA START     THE CONTINUATOR WILL BUMP THIS TO MAKE
0349  00056 072435R       STA CX05      A JUMP TO INITIATOR RETURN 0 (I.R.0)
0350  00057 026625R       JMP WRIT      SO WE CAN USE 'WRIT' AS A SUBROUTINE
0351  00060 000071R START DEF I.R.0-1
0352*
0353*
0354  00061 161667  READ  LDA EQT08,I   GET REQUEST LENGTH
0355  00062 002003        SZA,RSS       IS IT ZERO?
0356  00063 026073R       JMP I.R.4     YES, IMMEDIATE COMPLETION
0357*
0358  00064 161671        LDA EQT10,I   GET DESIRED #CHAR (-)
0359  00065 003004        CMA,INA       MAKE POSITIVE
0360  00066 002004        INA           PLUS 1
0361  00067 141670        ADA EQT09,I   ADD TO START OF BUFFER
0362  00070 172013R       STA EQX12,I   SAVE ADDRESS OF END OF BUFFER +1
0363  00071 017130R       JSB TRIGR     ISSUE TRIGGER IF NEEDED & START READ
0364*
0365*
0366  00072 006401  I.R.0 CLB,RSS       OPERATION INITIATED
0367  00073 066102R I.R.4 LDB K4        IMMEDIATE COMPLETION
0368  00074 017206R       JSB INTCN     CONFIGURE FOR INTERRUPT
0369  00075 060001        LDA BREG      GET CONDITION IN A REGISTER
0370                      IFN
0371                      LDB EQT06,I   ! B= CONTROL WORD
0372                      JSB TRAP      ! A= 0 OR 4
0373                      DEF *-ORG     ! S AS BEFORE
0374                      LDB AREG      RESTORE B REG
0375                      XIF
0376  00076 126015R       JMP IX05,I    RETURN TO RTIOC
0377*
0378  00077 017206R IXDWN JSB INTCN     RESET FOR INTERRUPTS
0379  00100 062027R       LDA K3        PRODUCE EQUIPMENT NOT READY MESSAGE
0380                      IFN
0381                      JSB TRAP
0382                      DEF *-ORG
0383                      XIF
0384  00101 126015R       JMP IX05,I
0385*
0386  00102         BIT2  EQU *
0387  00102 000004  K4    OCT 000004
0388  00103 177772  M6    OCT -6
0389  00104 030003  Z3.3  OCT 030003    DEFAULT FRAME CONTROL WORD
0390  01654         INTBA EQU 1654B
0391*
```

```
0392*
0393*  THIS IS ONE-SHOT CODE (FOR EACH EQT THAT IS) THAT SETS UP THE BREAK
0394                      IFN           !!!!!DEBUG
0395                      UNL
0396                      XIF           !!!!!
0397*  PROCESSING POINTERS.  THE ID SEGMENT OF THE BREAK PROCESSOR PROGRAM
0398*  IS MOVED FROM THE INTERRUPT TABLE WHERE IT WAS PLACED AT GEN' TIME
0399*  INTO THE EQT EXTENSION.  THE EQT ADDRESS OF THE INTERFACE IS THEN
0400*  PLACED IN THE INTERRUPT TABLE SO THAT NORMAL I/O PROCESSING CAN
0401*  BE DONE.
0402*  WE ALSO CHECK THE EQT EXTENSION LENGTH TO MAKE SURE ENOUGH WAS
0403*  PROVIDED.  IF WE DIDN'T DO THIS, SOME OF THE WEIRDEST BUGS EVER
0404*  SEEN COULD HAPPEN AT RUN TIME.
0405                      IFN           !!!!!DEBUG
0406                      LST
0407                      XIF           !!!!!
0408*
0409  00105 062430R SHOT1 LDA EQXML     SEE HOW MANY WORDS WERE GEN'D
0410  00106 141771        ADA EQT12,I   FOR THE EQT EXTENSION
0411  00107 002020        SSA           IF NOT ENOUGH,
0412  00110 102055        HLT 55B       STOP!!  PUSH RUN AT YOUR OWN RISK!
0413  00111 061654        LDA INTBA     GET ADDRESS OF INTERRUPT TABLE
0414  00112 042103R       ADA M6        LESS 6, 'CAUSE INTBL STARTS WITH 6
0415  00113 042373R       ADA S.C.      ADD CURRENT SELECT CODE
0416                      IFN
0417                      JSB TRAP
0418                      DEF *-ORG
0419                      JMP CCB       NO "PRMPT" ALLOWED FOR DEBUG
0420                      XIF
0421  00114 164000        LDB AREG,I    AND GET CONTENTS OF INTBL FOR THIS S.C.
0422  00115 007007        CMB,INB,SZB,RSS   MAKE POSITIVE, AND CHECK FOR ZERO
0423  00116 007400  CCB   CCB           IF ZERO, SAVE -1 INSTEAD
0424  00117 176001R       STB EQX02,I   PUT INTO EQT EXTENSION
0425                      IFN
0426                      JSB TRAP
0427                      DEF *-ORG
0428                      XIF
0429  00120 065660        LDB EQT01     GET THIS EQT'S ADDRESS
0430  00121 174000        STB AREG,I    AND PUT THAT IN THE INTERRUPT TABLE
0431*
0432  00122 161663        LDA EQT04,I   ALTER EQT 4 TO TELL EXEC
0433  00123 032176R       IOR BIT12     THAT WE HANDLE OUR OWN
0434  00124 171663        STA EQT04,I   TIME-OUTS
0435*
0436******   START CODE FOR PACED READS
0437  00125 062201R       LDA DC1       MAKE 'DC1' BE THE DEFAULT
0438  00126 172003R       STA EQX04,I   CHARACTER FOR READS
0439*******REQUEST CODE FOR PACED WRITES, REQUIRES
0440*******PERMISSION CODE ("ACK") FROM REMOTE DEVICE
0441  00127 062172R       LDA ENQ       *MAKE "ENQ" DEFAULT REQUEST CODE
0442  00130 172012R       STA EQX11,I   *AND STORE IT
0443                      IFN           !!!!!DEBUG
0444                      LDA =B77      LET "?" REPLACE "DC1"
0445                      STA EQX04,I
0446                      LDA =B43      LET "#" REPLACE "ENQ"
```

```
0447                    STA EQX11,I
0448                    XIF             !!!!!
0449  00131 062202R     LDA K6          MAKE 300 BAUD BE THE DEFAULT LINE
0450  00132 172000R     STA EQX01,I     SPEED
0451  00133 062104R     LDA Z3.3        DEFAULT FRAME CONTROL TO
0452  00134 172005R     STA EQX06,I     NO PARITY, 1 STOP BIT, 8 BIT CHAR.
0453* THE FOLLOWING CODE IS NEEDED BY PDP-HP INTERFACE
0454  00135 062027R     LDA K3          CODE FOR HANDSHAKE
0455  00136 172014R     STA EQX13,I     SETS BOTH ENT-ACK AND DC2-DC1
0456*
0457                    IFN             !!!!!DEBUG
0458                    UNL
0459                    XIF             !!!!!
0460* THE 12966 CARD HAS A 256 BY 1 RAM ON IT THAT IS USED TO DEFINE
0461* SPECIAL CHARACTERS WHICH INTERRUPT.  A CHARACTER IS USED AS THE
0462* ADDRESS INTO THE RAM AND THE STATE OF THE ADDRESSED DATA BIT INDICATES
0463* WHETHER THE CHARACTER IS A SPECIAL CHARACTER. (1==> SPECIAL)
0464* THE RAM IS VOLATILE SO ITS CONTENTS ARE LOST ON POWER FAILURE.  WHEN
0465* POWER IS RESTORED, THE RAM CONTENTS ARE RANDOM, SO IT MUST BE CLEARED
0466* BY WRITING ZERO TO ALL 256 LOCATIONS, AS THERE IS NO OVERALL CLEAR INPUT
0467* ON THE RAM.  THE INITIAL CLEAR IS DONE AS PART OF THE ONE-SHOT CODE, AND
0468* POWER FAIL CLEAR IS DONE ANY TIME THE INITIATOR IS ENTERED WITH THE FLAG
0469* ON THE 12966 SET.  WE THEN SET CARRIAGE RETURN AND RUB-OUT AS THE ONLY
0470* VALID TERMINATOR CHARACTERS.
0471                    IFN             !!!!!DEBUG
0472                    LST
0473                    XIF             !!!!!
0474* RAM CLEAR & SET
0475*
0476  00137 016203R CLR JSB MCLR        MUST BE IN XMIT MODE TO CLEAR RAM
0477  00140 066173R     LDB M377        USE B AS A COUNTER
0478  00141 062174R     LDA Z6C         TO SEND CLEAR 000 CHAR.
0479*
0480  00142 017202R CLOOP JSB OTA       CLEAR A CHAR. IN RAM
0481  00143 002004      INA             MAKE 001, THEN 002, ETC.
0482  00144 006006      INB,SZB         FOR CHAR. 000 THROUGH 377
0483  00145 026142R     JMP CLOOP
0484*
0485  00146 000000  SPTBL NOP
0486  00147 066171R     LDB XSPTL       * LOAD LENGTH OF SPECIAL CH. TABLE
0487  00150 007004      CMB,INB         * MAKE IT NEGATIVE
0488  00151 062164R     LDA .SPT        * GET ADDR OF TABLE
0489  00152 072547R     STA FLAG        * STORE IT AS A POINTER
0490  00153 162547R NEXT1 LDA FLAG,I    * GET SPECIAL CHAR FROM TABLE
0491  00154 042175R     ADA Z6S         * CONTROL WORD TO SET SPECIAL CH. RAM
0492  00155 017202R     JSB OTA         * OUTPUT TO RAM
0493  00156 032455R     IOR BIT7        * SET PARITY BIT
0494  00157 017202R     JSB OTA         * OUTPUT TO RAM
0495  00160 036547R     ISZ FLAG        * NEXT SPECIAL CHAR ADDRESS
0496  00161 006006      INB,SZB         * INCREMENT BREG, IS IT ZERO?
0497  00162 026153R     JMP NEXT1       * NO,DO NEXT CH.
0498                    IFN
0499                    JSB TRAP
0500                    DEF *-ORG
0501                    XIF
```

```
0502  00163 026024R       JMP RFLGS    * YES,REJOIN INITIATOR
0503*
0504  00164 000165R .SPT   DEF *+1
0505  00165 000015  CR     OCT 15        * CR (BEGIN SPECIAL CHARS TABLE)
0506  00166 000177  DEL    OCT 177       * DEL
0507  00167 000006  ACK    OCT 06        * ACK
0508  00170 000022  DC2    OCT 22        * DC2
0509* ADD ANY OTHER SPECIAL CHARACTERS HERE
0510  00171 000004  XSPTL DEF *-.SPT-1  * LENGTH OF SPECIAL CHARACTERS TABLE
0511*
0512*
0513  00172 000005  ENQ    OCT 000005
0514*
0515  00173 177401  M377   OCT -377
0516  00174 060000  Z6C    OCT 060000    CONTROL WORD TO CLEAR RAM
0517  00175 060400  Z6S    OCT 060400    CONTROL WORD TO SET RAM
0518  00176 010000  BIT12  OCT 010000
0519  00177 060415  SP15   OCT 060415
0520  00200 060577  SP177  OCT 060577
0521  00201 000021  DC1    OCT 21        ASCII DC1 CHARACTER
0522  00202 000006  K6     OCT 6
0523*
0524*
0525*  MCLR INITIALIZES THE 12966 CARD TO A KNOWN STATE.
0526*  IT MUST BE CALLED WHENEVER THE 'FIFO RIPPLE THROUGH TRICK' IS
0527*  USED SO THAT BUFFER EMPTY WILL AGAIN BE RECOGINIZED.
0528*
0529*  INITIAL CONDITIONS:
0530*
0531*  WORD 0: NOT SENT
0532*  WORD 1: INTERRUPT MASK = CHECK FOR CB,CC,CF
0533*  WORD 2: LAST STATUS PER EQX10
0534*  WORD 3: CONTENTS OF EQT EXTENSION WORD 6
0535*  WORD 4: XMIT, CA,CD ON, BAUD PER EQX01
0536*  WORD 5: CLEAR ALL INTERRUPT STATUS FLIP-FLOPS
0537*  WORD 6: NOT SENT
0538*
0539* WORD 4 IS USED AS THE MASTER RESET WORD, FOR IF ANY OTHER WORD IS
0540* USED, THE CA & CD BITS WOULD GO FALSE, WHICH WILL MAKE THE MODEM
0541* HANG UP THE PHONE.
0542*
0543*
0544  00203 000000  MCLR   NOP          THIS ROUTINE RESETS THE CARD
0545  00204 162000R       LDA EQX01,I   GET BAUD SETTING
0546                      IFN           !!!!!DEBUG
0547                      OTA 1         ! BAUD SETTING TO S REG
0548                      XIF           !!!!!
0549  00205 032217R       IOR Z4MR      GET MASTER RESET, XMIT, CA,CD ON.
0550  00206 017202R       JSB OTA       ZAP THE CARD
0551  00207 063170R       LDA Z1        RESET THE INTERRUPT MASK
0552  00210 017202R       JSB OTA
0553  00211 162011R       LDA EQX10,I   GET LAST LINE STATUS
0554                      IFN           !!!!!DEBUG
0555                      LDB AREG      ! STATUS TO BREG
0556                      XIF           !!!!!
```

```
0557  00212 032220R        IOR Z2         MAKE CONTROL WORD TO
0558  00213 017202R        JSB OTA        SET DEVICE REFERENCE REGISTER
0559  00214 162005R        LDA EQX06,I    GET CURRENT FRAME SETTINGS
0560  00215 017202R        JSB OTA
0561                       IFN
0562                       JSB TRAP      ! A= FRAME SETTINGS, EQX06
0563                       DEF *-ORG     ! B= STATUS FROM EQX10
0564                       XIF           ! S= CURRENT BAUD FROM EQX01
0565  00216 126203R        JMP MCLR,I
0566*
0567  00217 140700  Z4MR   OCT 140700     MASTER RESET WORD
0568  00220         BIT13  EQU *
0569  00220 020000  Z2     OCT 020000
0570  00221 000037  K37    OCT 000037



0572*  THIS SUBROUTINE CHECKS FOR BITS BEING SET IN THE CONTROL WORD
0573*  JSB CNWD          RETURNS WITH A=IAND(EQT06,MASK)
0574*  OCT XX            MASK TO USE
0575*  ( RETURN IF NO BITS SET
0576*  ( RETURN IF ANY BIT SET
0577  00222 000000  CNWD   NOP
0578  00223 161665         LDA EQT06,I    GET THE CONTROL WORD
0579                       IFN
0580                       OTB 1          SAVE BREG
0581                       LDB CNWD,I     ! A= CONTROL WORD,EQT06
0582                       JSB TRAP       ! B=USER MASK
0583                       DEF *-ORG      ! S=PREVIOUS B VALUE
0584                       LIB 1           RESTORE B REG
0585                       XIF
0586  00224 112222R        AND CNWD,I     AND IT WITH USERS MASK
0587  00225 036222R        ISZ CNWD       PUSH RETURN TO P+2
0588  00226 002002         SZA            IS RESULT ZERO?
0589  00227 036222R        ISZ CNWD       NO, RETURN P+3
0590  00230 126222R        JMP CNWD,I
```

```
0592  00231 161665  CNTRL LDA EQT06,I   GET THE CONWD AGAIN
0593  00232 101046        LSR 6         FUNC. IN BITS 10-6 TO BITS 4-0
0594  00233 012221R       AND K37
0595                      IFN
0596                      JSB TRAP
0597                      DEF *-ORG
0598                      XIF
0599  00234 042236R       ADA TBL       ADD START OF TABLE
0600  00235 124000        JMP AREG,I    AND VECTOR THROUGH THE TABLE
0601*
0602  00236 100237R TBL   DEF *+1,I     CONTROL CODE JUMP TABLE
0603  00237 000366R       DEF CN00       0 SYSTEM CLEAR
0604  00240 000277R       DEF RJCT2      1 WRITE EOF
0605  00241 000277R       DEF RJCT2      2 BACKSPACE RECORD
0606  00242 000277R       DEF RJCT2      3 FORWARD SPACE RECORD
0607  00243 000277R       DEF RJCT2      4 REWIND
0608  00244 000277R       DEF RJCT2      5 REWIND
0609  00245 000277R       DEF RJCT2      6 DYNAMIC STATUS
0610  00246 000277R       DEF RJCT2      7 CONDITIONAL T.O.F. ?
0611  00247 000277R       DEF RJCT2     10 WRITE EOF / PRODUCE LEADER
0612  00250 000073R       DEF I.R.4     11 LINE SPACING
0613  00251 000277R       DEF RJCT2     12
0614  00252 000277R       DEF RJCT2     13 FORWARD SPACE FILE
0615  00253 000277R       DEF RJCT2     14 BACK SPACE FILE
0616  00254 000277R       DEF RJCT2     15
0617  00255 000277R       DEF RJCT2     16
0618  00256 000277R       DEF RJCT2     17
0619  00257 000303R       DEF CN20      20 ENABLE TERMINAL
0620  00260 000307R       DEF CN21      21 DISABLE TERMINAL
0621  00261 000312R       DEF CN22      22 SET TIME OUT
0622  00262 000277R       DEF RJCT2     23 BUFFER FLUSH
0623  00263 000277R       DEF RJCT2     24 RESTORE OUTPUT PROCESSING
0624  00264 000316R       DEF CN25      25 UPDATE TERMINAL HANDSHAKING PROTOCOL
0625  00265 000277R       DEF RJCT2     26 WRITE EOD
0626  00266 000277R       DEF RJCT2     27 LOCATE FILE
0627  00267 000344R       DEF CN30      30 INITIALIZE LINE   SET BAUD & CHAR. FR
0628  00270 000337R       DEF CN31      31 LINE CONTROL - HANG UP ON MA BELL
0629  00271 000325R       DEF CN32      32 LINE CONTROL - AUTO ANSWER
0630*     DEF CN33      33 SPEED SENSE
0631  00272 000277R       DEF RJCT2     33 NO SPEED SENSE
0632  00273 000277R       DEF RJCT2     34 SET TERMINATOR BYTE FOR READS
0633  00274 000277R       DEF RJCT2     35
0634  00275 000321R       DEF CN36      36 SET CARRIAGE RETURN/LINE FEED DELAY
0635  00276 000277R       DEF RJCT2     37 RESERVED CODE FOR RTE-L DISC DRIVERS



0637  00277 062301R RJCT2 LDA K2        SET A=2, ILLEGAL CONTROL
0638  00300 126015R       JMP IX05,I    RETURN
0639  00301         BIT1  EQU *
0640  00301 000002  K2    OCT 000002
```

```
0642  00302 177677  NBIT6 OCT 177677
0643*
0644  00303 017114R CN20  JSB SSTAT     SET BIT IN EQT05
0645  00304 000002        OCT 000002    TO SAY TEMINAL IS ENABLED
0646  00305 026367R       JMP RSET      MAKE SURE BREAK AND C.R. CAN INTERRUPT
0647  00306 026073R       JMP I.R.4
0648*
0649  00307 017122R CN21  JSB CSTAT     CLEAR BIT IN EQT05
0650  00310 177775  M2    OCT 177775    TO SAY TERMINAL DISABLED
0651  00311 026073R       JMP I.R.4     IMMEDIATE COMPLETION
0652*
0653  00312 161666  CN22  LDA EQT07,I   GET NEW TIME-OUT FROM
0654  00313 003004        CMA,INA       USER, NEGATE IT AND
0655  00314 171773        STA EQT14,I   PUT IT IN THE EQT
0656  00315 026367R       JMP RSET      SET TO RCV AND LEAVE
0657*
0658*
0659  00316 161666  CN25  LDA EQT07,I   GET HANDSHAKE BITS (DEFAULTS TO BOTH ON)
0660  00317 172014R       STA EQX13,I   KEEP FOR "HAND" ROUTINE
0661  00320 026073R       JMP I.R.4     IMMEDIATE RETURN
0662*
0663*



0665  00321 161666  CN36  LDA EQT07,I   GET THE NEW C.R./L.F. DELAY FROM USER
0666  00322 003004        CMA,INA       LET THE USER USE POS. NUMBERS
0667  00323 172006R       STA EQX07,I   PUT INTO EQT EXTENSION
0668  00324 026367R       JMP RSET      SET TO RCV AND LEAVE



0670  00325 062220R CN32  LDA Z2        SET DEVICE STATUS REFERENCE
0671  00326 017202R       JSB OTA       TO ALL 0'S
0672  00327 062336R       LDA Z1RNG     SET THE MASK REGISTER TO INTERRUPT
0673  00330 017202R       JSB OTA       ON RING INDICATOR
0674  00331 002400        CLA           CLEAR THE TIME-OUT ON THIS
0675  00332 171774        STA EQT15,I   EQT TO WAIT FOR PHONE TO RING
0676  00333 017114R       JSB SSTAT     SET 'WAITING FOR RING'
0677  00334 000100        OCT 000100    BIT IN STATUS WORD (EQT05)
0678  00335 026072R       JMP I.R.0     TELL EXEC WE STARTED IT
0679*
0680  00336 010004  Z1RNG OCT 010004


0682  00337 062447R CN31  LDA Z4CLR     GET A CONTROL WORD TO SET
0683  00340 132000R       IOR EQX01,I   CA & CD FALSE, INCLUDE BAUD,
0684  00341 017202R       JSB OTA       AND SET CARD
0685  00342 026073R       JMP I.R.4     DONE


0687*CN33 JSB SSTAT     SET THE 'SPEED SENSING' BIT
0688*     OCT 000004
0689*     LDA EQX01,I   GET THE BAUD CONFIGURATION
0690*     IOR K17       FORCE TO 9600 BAUD
```

```
0691*     STA EQX01,I   AND PUT BACK
0692*     LDA Z1CM      SET FOR SINGLE CHARACTER MODE
0693*     JSB OTA
0694*     JSB RCV       AND SET INTO RECEIVE
0695*     JMP I.R.0     TELL EXEC IT IS GOING
0696*
0697*Z1CM OCT 010072
0698*K17  OCT 000017



0700                    IFN           !!!!!DEBUG
0701                    UNL
0702                    XIF           !!!!!
0703*   BIT USAGE IN EQT07 FOR CN30 CALL:
0704*         X HCC XXX LSX PPB BBB
0705*  PP ==> 0X FOR NO PARITY CHECKING
0706*     ==> 10 FOR ODD PARITY
0707*     ==> 11 FOR EVEN PARITY
0708*  L  ==> 0 FOR HARDWIRED LINE,  1 FOR MODEM CONNECTED
0709*         (DEFINITION USED BY DVA05, DVX05 DOES NOT USE.)
0710*  S  ==> 0 FOR ONE STOP BIT, 1 FOR 2 STOP BITS (1.5 @ 5 BITS/CHAR.)
0711*  CC ==> 00 FOR 8 DATA BITS, 01 FOR 7, 10 FOR 6, AND 11 FOR 5.
0712*         NOTE! WHEN PARITY IS ENABLED, 1 IS SUBTRACTED
0713*         FROM THE NUMBER OF DATA BITS. (WITH ROLL-AROUND FROM 5 TO 8)
0714*  H  ==> 1 TO ALWAYS INHIBIT ECHO, OVERRIDING BIT 8 OF THE CONWD.
0715*
0716*  B BBB = BAUD SETTING::
0717*
0718*BAUD EXT 50 75 110 135 150 300 600 900 1200 1800 2400 3600 4800 7200 9600
0719*CODE  00 01 02  03  04  05  06  07  10   11   12   13   14   15   16   17
0720*
0721* THE TABLE GIVEN ABOVE ASSUMES THAT PIN J1-K IS STRAPPED TO J1-H IN THE
0722*  CONNECTOR HOOD.
0723*  IF J1-K IS MOVED TO J1-L, HALVE THE ABOVE BAUD SETTING ( 4800 BAUD MAX)
0724*  IF J1-K IS MOVED TO J1-J, DOUBLE THE ABOVE BAUD SETTING  (19.2K MAX)
0725*  IF J1-K IS MOVED TO J1-M, QUADRUPLE THE ABOVE BAUD SETTING  (38.4K MAX)
0726*  IF J1-K IS MOVED TO J1-B, OCTUPLE THE ABOVE BAUD SETTING    (76.8K MAX)
0727* NOTE!!!!!   HP ONLY SUPPORTS J1-K CONNECTIONS, BUT THE OTHERS MIGHT
0728* WORK.  THE RATES ABOVE 19.2 MIGHT BE BANDWIDTH LIMITED BY THE LINE DRIVERS
0729* USED ON THE CARD (PER RS232 SPEC.) OR THE MAXIMUM RATE OF THE UART.
0730* THE SPECIALS GROUP AT DSD OFFERS A MODIFIED CARD WHICH IS CAPABLE OF
0731* APPROX. 254K BAUD. CONTACT YOUR LOCAL SALES OFFICE.


0733* SOME EXAMPLES MIGHT HELP HERE:
0734* TO RUN MOST CRT'S @ 9600 BAUD WOULD REQUIRE ) CN,LU,30B,000417B
0735*  WHICH WOULD BE ' ST D1 D2 D3 D4 D5 D6 D7 D8 SP '
0736*  THIS IS A START BIT, FOLLOWED BY 8 DATA BITS, AND A SINGLE STOP BIT.
0737* TO RUN AN ASR33 TELETYPE @ 110 BAUD, NO PARITY WOULD REQUIRE
0738*  CN,LU,30B,004403B ==> ' ST D1 D2 D3 D4 D5 D6 D7 D8 SP SP '
0739* A CRT @ 2400 BAUD WITH EVEN PARITY IS  CN,LU,30B,000473B
0740*  WHICH WOULD BE ' ST D1 D2 D3 D4 D5 D6 D7 PA SP '
0741* A BAR CODE READER @ 4800 BAUD, EVEN PARITY, WHICH DOES NOT NEED ECHO
0742*  WOULD BE CN,LU,30B,040475B ==) ' ST D1 D2 D3 D4 D5 D6 D7 PA SP '
```

```
0743* SOME DEVICES, SUCH AS THE MOTOROLA AART'S, NEED 8 DATA BITS PLUS PARITY
0744*  WHICH IS AN 11 BIT CHAR.  THE SETTING FOR THIS IS CN,LU,30B,003451B
0745*  AT 1200 BAUD.  THIS GIVES ==) ' ST D1 D2 D3 D4 D5 D6 D7 D8 PA SP '


0747                      IFN           !!!!!DEBUG
0748                      LST
0749                      XIF           !!!!!
0750  00343 040017  K4.17 OCT 040017
0751*
0752  00344 161666  CN30  LDA EQT07,I   GET THE CONFIGURATION WORD
0753  00345 012343R       AND K4.17     MASK TO BAUD AND DUPLEX BITS
0754  00346 172000R       STA EQX01,I   SAVE IN EQT EXTENSION WORD 1
0755                      IFN           !!!!!DEBUG
0756                      OTA 1         TO S REG FOR DEBUG
0757                      XIF           !!!!!
0758*
0759  00347 161666        LDA EQT07,I   GET THE WORD AGAIN  X HCC XXX LSX PPB BBB
0760  00350 012371R       AND K260      MASK PARITY, #STOPS 0 000 000 0S0 PP0 000
0761  00351 001727        ALF,ALF       MOVE PARITY ENABLE  S 0PP 000 000 000 000
0762  00352 001222        RAL,RAL       TO SIGN BIT         P P00 000 000 000 0S0
0763  00353 064000        LDB AREG      SAVE IN A SAFE PLACE
0764  00354 162006R       LDA EQX07,I   GET CONFIG. AGAIN   X HCC XXX LSX PPB BBB
0765  00355 001700        ALF           SIZE TO LOW BITS    X XXL SXP PBB BBX HCC
0766  00356 006020        SSB           IF PARITY ENABLED,
0767  00357 002004        INA           SIZE=SIZE-1         X XXL SXP PBB BBX HCC
0768  00360 003000        CMA           REVERSE BITS        X XX1 sXp pbb bbX hcc
0769  00361 012027R       AND K3        MASK TO SIZE BITS   0 000 000 000 000 0cc
0770  00362 005700        BLF           MOVE PARITY & #STOP 0 000 000 000 S0P P00
0771  00363 030001        IOR BREG      MERGE CONTROL BITS  0 000 000 000 S0P Pcc
0772  00364 032372R       IOR Z3        MAKE CONTROL WORD   0 011 000 000 S0P Pcc
0773  00365 172005R       STA EQX06,I   AND SAVE IT
0774                      IFN
0775                      JSB TRAP
0776                      DEF *-ORG
0777                      XIF
0778  00366 016203R CN00  JSB MCLR      RE-CONFIGURE THE CARD & MASTER RESET
0779  00367 017223R RSET  JSB RCV       RCV MODE ALLOWS BREAK AND C.R. TO INTERR
0780*                   IN THIS VERSION (SEE CONTINUATION SECTION)
0781  00370 026073R       JMP I.R.4     IMMEDIATE COMPLETION
0782*
0783  00371 000260  K260  OCT 000260
0784  00372 030000  Z3    OCT 030000
```

```
0786  00373 000000  S.C.  NOP
0787*
0788*
0789  00374 000000  SETIO NOP
0790                       IFN           !!!!!DEBUG
0791                       JMP *+3       ! SET IO EACH TIME
0792                       XIF           !!!!!
0793  00375 052373R        CPA S.C.      SEE IF ALREADY CONFIGURED
0794  00376 126374R        JMP SETIO,I   YES, SO NO NEED TO DO IT AGAIN
0795*
0796  00377 072373R        STA S.C.      SAVE FOR NEXT TIME
0797  00400 032434R        IOR SFC       MAKE AN SFC INSTR.
0798  00401 072022R        STA SFC1
0799  00402 042432R        ADA K300      MAKE INTO LIA INSTR.
0800  00403 073177R        STA LIA1
0801  00404 072772R        STA LIA2
0802  00405 042444R        ADA BIT6      MAKE INTO OTA INSTR.
0803  00406 072650R        STA OTA1
0804  00407 073203R        STA OTA2
0805  00410 042444R        ADA BIT6      MAKE STC
0806  00411 073174R        STA STC1
0807  00412 042460R        ADA BIT9      MAKE STC SC,C
0808  00413 073211R        STA STCC1
0809  00414 042433R        ADA K3000     FINALLY, A CLC.
0810  00415 073172R        STA CLC1
0811*     STA CLC2
0812*
0813  00416 161772         LDA EQT13,I   GET ADDRESS OF EQT EXTENSION
0814  00417 066431R        LDB EQXAD     GET ADDRESS OF LOCAL POINTERS
0815*
0816                       IFN           !!!!!DEBUG
0817                       UNL
0818                       XIF           !!!!!
0819* IF $SETP AVAILABLE IN MICROCODE (PART OF THE FFP FIRMWARE),
0820*   THEN --------
0821*
0822*     EXT $SETP
0823*     JSB $SETP     USE THE FFP MICROCODE IF AVAILABLE
0824*     ABS 105227B   <<<<<  ABSOLUTE VERSION ONLY!!!($SETP)   >>>
0825*     DEF EQX0L     DEF TO NUMBER OF EQT EXTENSION WORDS
0826*
0827*  ELSE --------
0828                       IFN           !!!!!DEBUG
0829                       LST
0830                       XIF           !!!!!
0831  00420 076547R        STB FLAG      SAVE POINTER TO LOCAL AREA
0832  00421 066430R        LDB EQXML     GET NUMBER OF POINTERS
0833  00422 172547R STLP   STA FLAG,I    STORE ONE ADDRESS INTO A POINTER.
0834  00423 002004         INA           NEXT ADDRESS
0835  00424 036547R        ISZ FLAG      NEXT POINTER
0836  00425 006006         INB,SZB       DONE?
0837  00426 026422R        JMP STLP      NO, DO MORE
0838*
0839* END ------
0840*
```

```
0841  00427 126374R       JMP SETIO,I   RETURN, CONFIGURATION COMPLETE
0842*
0843* DEF FOR FFP VERSION ONLY
0844*EQX0L ABS XEQTL
0845* DEF FOR NON-FFP VERSION ONLY
0846  00430 177763  EQXML ABS -XEQTL
0847  00431 000000R EQXAD DEF EQX01
0848  00432 000300  K300  OCT 000300
0849  00433 003000  K3000 OCT 003000
0850  00434 102200  SFC   SFC 00
```

```
0852* IN THIS VERSION, C.R. CAN GENERATE AN UNEXPECTED INTERRUPT
0853* THAT IS EQUIVALENT TO "BREAK" IF THE BOARD WAS LEFT IN RECEIVE
0854* MODE BY PREVIOUS CALLS. "BREAK" ALWAYS FUNCTIONS WHETHER THE
0855* BOARD IS IN RCV OR XMIT MODE.
0856  00435 000000  CX05  NOP           ENTRY TO CONTINUATOR
0857  00436 016374R       JSB SETIO     CONFIGURE I/O INSTRUCTIONS
0858  00437 017171R RESTA JSB STATS     GO READ CARD STATUS
0859*
0860*
0861                      IFN
0862                      JSB TRAP
0863                      DEF *-ORG
0864                      XIF
0865  00440 162007R       LDA EQX08,I   GET STATUS BACK
0866  00441 002020        SSA           IS SIGN BIT SET?
0867  00442 026556R       JMP LINE      YES, DEVICE RS232 LINES HAVE CHANGED
0868*
0869  00443 016547R       JSB FLAG
0870  00444 000100  BIT6  OCT 000100    BREAK FLAG?
0871  00445 027260R       JMP BREAK
0872*
0873  00446 016547R       JSB FLAG
0874  00447         Z4CLR EQU *
0875  00447 040000  BIT14 OCT 040000    SPECIAL CHAR.
0876  00450 026474R       JMP SPC!
0877*
0878  00451 161660        LDA EQT01,I   GET THE LIST LINKAGE WORD
0879  00452 002003        SZA,RSS       IF IT IS ZERO WE HAVE NO CALL IN
0880  00453 026753R       JMP CEXIT     PROGRESS, SO DEPART
0881*
0882  00454 016547R       JSB FLAG
0883  00455 000200  BIT7  OCT 000200    BUFFER EMPTY
0884  00456 026523R       JMP BFMPT
0885*
0886  00457 016547R       JSB FLAG
0887  00460 001000  BIT9  OCT 001000    BUFFER HALF FULL
0888  00461 026757R       JMP INPUT
0889*
0890*
0891  00462 016547R       JSB FLAG      CHECK FOR OVERRUN/PARITY ERROR
0892  00463         BLANK EQU *
0893  00463 000040  BIT5  OCT 000040
0894  00464 026541R       JMP OPERR     WE HAVE AN OVERRUN OR PARITY ERROR
0895*
0896  00465 017242R       JSB SUBCH     TEST EQT04 FOR TIME-OUT
0897  00466 004000  BIT11 OCT 004000    BIT 11 SET?
0898  00467 026516R       JMP T.OUT     YES, HANDLE TIME-OUT
0899*
0900  00470 016547R       JSB FLAG
0901  00471 000400        OCT 000400    BUFFER FULL?
0902  00472 026757R       JMP INPUT     HANDLE JUST LIKE A HALF FULL
0903*
0904                      IFN           !!!!!DEBUG
0905                      UNL
0906                      XIF           !!!!!
```

```
0907*     HLT 67B       << FOR DEBUG
0908*     LDA EQT05,I   MUST BE A SINGLE CHAR. INTERRUPT
0909*     AND BIT2      CAUSE THATS ALL THATS LEFT
0910*     SZA,RSS       ARE WE ATTEMPTING TO SPEED SENSE?
0911*
0912  00473 026753R       JMP CEXIT     UNKNOWN INTERRUPT, IGNORE IT
0913*
0914*CLC2 CLC 00        CLEAR THE CARD SO IT CAN'T INTERRUPT
0915*     LDA BIT5      SET A DELAY OF 400MS
0916*     JSB T.O.I     TO GET VERY SLOW CHARACTERS
0917*     JSB LIA       NOW, SEE HOW MANY CAME IN
0918*     HLT 66B       << FOR DEBUG
0919*
0920*     JSB MCLR      CLEAR THE BUFFER OF DEBRIS
0921*     JSB CSTAT     CLEAR THE "SPEED SENSE" FLAG
0922*     OCT 177773
0923*     JMP CMPLT     TELL EXEC IT IS ALL DONE
0924*
0925                     IFN           !!!!!DEBUG
0926                     LST
0927                     XIF           !!!!!
0928*
0929  00474 000000  SPC! NOP
0930                     IFN           !!!!DEBUG
0931                     JSB TRAP
0932                     DEF *-ORG
0933                     XIF           !!!!
0934  00475 161660       LDA EQT01,I   GET LINK LIST WORD
0935  00476 002003       SZA,RSS       IF NO CALL IS IN PROGRESS
0936  00477 026504R      JMP CHECK     CHECK FOR BREAK
0937  00500 016222R      JSB CNWD      TEST THE CONTROL WORD
0938  00501 000001       OCT 000001    IF A READ IS IN PROGRESS
0939  00502 002001       RSS
0940  00503 026757R      JMP INPUT     HANDLE SPECIAL CHARS THERE
0941*
0942* OTHERWISE A WRITE IS IN PROGRESS OR THIS IS AN UNEXPECTED INTERRUPT
0943*
0944  00504 017176R CHECK JSB LIA       FIND OUT WHICH SPECIAL CHAR WAS RECEIVED
0945  00505 013021R       AND K177      MASK OUT COUNTER BITS AND PARITY BIT
0946                      IFN
0947                      JSB TRAP
0948                      DEF *-ORG
0949                      XIF
0950  00506 052515R       CPA B.C.      IS THIS A BREAK CHAR (B.C. NORMALLY = CR
0951  00507 027260R       JMP BREAK     YES,  GET SYSTEM PROMPT IF ENABLED
0952  00510 052167R       CPA ACK       THIS MAY BE "OK TO SEND" FROM TERMINAL
0953  00511 027077R       JMP ACK!      NORMAL FOR OUTPUT HANDSHAKE (WRITE ONLY)
0954  00512 052201R       CPA DC1       USER WANTS TO TRANSMIT??
0955  00513 027260R       JMP BREAK     END THE WRITE(IF ANY) AND LET HIM DO SO
0956  00514 026753R       JMP CEXIT     ILLEGAL CHAR, GO AWAY
0957*
0958  00515 000015  B.C.  OCT 000015    BREAK CHAR = C.R. BY DEFAULT
0959*
0960  00516 161664  T.OUT LDA EQT05,I   SEE IF TIME OUT WAS EXPECTED.
0961                      IFN
```

```
0962                    JSB TRAP
0963                    DEF *-ORG
0964                    XIF
0965  00517 002011      SLA,RSS       BY CHECKING LSB OF STATUS WORD
0966  00520 026531R     JMP TOUT?     NO, THIS IS AN UNEXPECTED TIME-OUT
0967*
0968  00521 017122R     JSB CSTAT     YES, CLEAR THE BIT FOR NEXT TIME
0969  00522 177776      OCT 177776    ALL EXCEPT BIT 0
0970  00523 161672 BFMPT LDA EQT11,I   GET THE CO-ROUTINE ADDRESS
0971                    IFN
0972                    JSB TRAP
0973                    DEF *-ORG
0974                    XIF
0975  00524 105774      CBS Mi EQT11,I  CLEAR ADDRESS TO PREVENT RE-USE
      00525 001026R
      00526 101672
0976  00527 124000      JMP AREG,I    JUMP TO CO-ROUTINE


0978  00530 016203R BRKRT JSB MCLR
0979  00531 006400  TOUT? CLB          RETURN A TRANSMISSION LOG OF ZERO
0980  00532 161660      LDA EQT01,I   ARE THERE ANY OUTSTANDING
0981                    IFN
0982                    JSB TRAP
0983                    DEF *-ORG
0984                    XIF
0985  00533 002002      SZA           TRANSFERS?
0986  00534 026721R     JMP CMPLT     YES, COMPLETE ANY TRANSFER IN PROGRESS
0987  00535 026753R     JMP CEXIT     IGNORE IT, MAYBE IT WILL GO AWAY
0988*
0989  00536 017206R CXDWN JSB INTCN     RESET FOR INTERRUPTS
0990  00537 002404      CLA,INA       I/O NR En Ln Sn  (A=1)
0991  00540 126435R     JMP CX05,I
0992*
0993*
0994  00541 017114R OPERR JSB SSTAT     SET ERROR BIT IN STATUS WORD
0995  00542 000010  BIT3  OCT 000010
0996  00543 062546R     LDA ZPCLR     THEN GO CLEAR THE OVERRUN/PARITY
0997  00544 017202R     JSB OTA       FLIP-FLOP
0998  00545 026437R     JMP RESTA     THEN REDO THE STATUS CHECKS
0999*
1000  00546 050001  ZPCLR OCT 050001


1002*  THIS ROUTINE CHECKS THE LAST WORD READ FROM THE CARD FOR THE
1003*  PRESENCE OF THE BITS INDICATED IN THE CALL.
1004*    JSB FLAG
1005*    OCT XXXXXX    WORD TO USE AS A MASK
1006*    (RETURN IF ANY BIT SET)
1007*    (RETURN IF NO BIT IS SET)
1008*    ON RETURN,  A=IAND(EQX08,MASK)
1009*    B,E,O,X,Y UNALTERED
1010*
1011  00547 000000  FLAG  NOP
1012  00550 162547R     LDA FLAG,I    GET THE MASK FROM CALLER
```

```
1013  00551 036547R       ISZ FLAG
1014  00552 112007R       AND EQX08,I   AND WITH STATUS
1015                      IFN
1016                      JSB TRAP
1017                      DEF *-ORG
1018                      XIF
1019  00553 002003       SZA,RSS       WERE ANY OF THE INDICATED BITS SET?
1020  00554 036547R       ISZ FLAG      NO, RETURN PAST THE VECTOR
1021  00555 126547R       JMP FLAG,I
```

```
1023* THIS SECTION HANDLES CHANGES IN THE RS232 LINES.
1024*
1025  00556 012102R LINE  AND BIT2      STATUS IS IN A, SO SEE IF
1026  00557 002002        SZA           PHONE IS RINGING.
1027  00560 026577R       JMP RING      YES, ANSWER IT
1028  00561 017156R       JSB LINCK     CHECK THE STATUS OF THE LINES
1029  00562 026613R       JMP LINDN     NOT GOOD, BOMB OUT
1030*
1031  00563 161664        LDA EQT05,I   LINES ARE UP, GET STATUS
1032  00564 012444R       AND BIT6      MASK TO 'WAITING FOR PHONE TO RING'
1033  00565 002003        SZA,RSS       IS IT SET?
1034  00566 026572R       JMP LINUP     NO, NORMAL LINE UP PROCESSING
1035*
1036  00567 121664        XOR EQT05,I   YES, REMOVE WAITING BIT
1037  00570 171664        STA EQT05,I   AND PUT THE STATUS WORD BACK
1038  00571 026721R       JMP CMPLT     PHONE HAS BEEN ANSWERED.
1039*
1040  00572 017122R LINUP JSB CSTAT     CLEAR THE 'DATA SET NOT READY'
1041  00573 177737        OCT 177737    IN BIT 4 OF THE STATUS WORD
1042  00574 171664        STA EQT05,I   IN THE STATUS WORD
1043  00575 017206R       JSB INTCN     CLEAR ALL FLAGS
1044  00576 026001X       JMP $UPIO     TELL EXEC TO UP THIS EQT
1045*     JMP LINK1,I
1046*LINK1 DEF $UPIO
1047*
1048*
1049  00577 161664  RING  LDA EQT05,I   GET STATUS WORD
1050  00600 012444R       AND BIT6      MASK TO 'WAITING FOR PHONE TO RING'
1051  00601 002003        SZA,RSS       IS IT SET?
1052  00602 026753R       JMP CEXIT     NO, IGNORE THE INTERRUPT
1053  00603 017223R       JSB RCV       YES, SET CA,CD BITS & BAUD
1054  00604 063170R       LDA Z1        AND ENABLE CB,CC & CF
1055  00605 017202R       JSB OTA       TO INTERRUPT AGAIN
1056*
1057*  THIS SECTION CHECKS THE LINE ONCE EVERY 2 SECONDS UNTIL
1058*  THE MODEM COMES ON LINE.
1059*
1060  00606 062622R RING1 LDA M200      NOW WE WAIT FOR THE
1061  00607 016742R       JSB T.O.I       DATA SET TO COME ON LINE
1062  00610 017156R       JSB LINCK     DID IT?
1063  00611 026606R       JMP RING1     NO, WAIT AGAIN
1064  00612 026721R       JMP CMPLT     YES, END OF REQUEST
1065*
1066*
1067  00613 161664  LINDN LDA EQT05,I   GET STATUS
1068  00614 012444R       AND BIT6      MASK TO WAITING FOR RING
1069  00615 002002        SZA
1070  00616 026606R       JMP RING1     YES, CONTINUE TO WAIT!!
1071  00617 017114R       JSB SSTAT     SET THE 'DATA SET DOWN' BIT
1072  00620 000020  BIT4  OCT 000020    IN THE STATUS WORD
1073  00621 026536R       JMP CXDWN     AND TELL RTE ABOUT IT TOO.
1074*
1075  00622 177470  M200  DEC -200      200 CENTI-SECONDS = 2 SEC.
```

```
1077  00623 000100  MAXFR DEC 64        HALF THE BUFFER LENGTH
1078  00624 177700  NMXFR DEC -64
1079*
1080*
1081  00625         WRIT  EQU *
1082  00625 165671  WRIT1 LDB EQT10,I   GET THE CHARACTER COUNT (-#)
1083                      IFN
1084                      JSB TRAP
1085                      DEF *-ORG
1086                      XIF
1087  00626 006003        SZB,RSS       IS IT 0?
1088  00627 026711R       JMP CRLF      YES, SKIP A LINE
1089  00630 046623R       ADB MAXFR     ADD MAX# OF CHAR IN A BURST
1090  00631 006021        SSB,RSS       SEE IF ANY WOULD BE LEFT IN BUFFER
1091  00632 026637R       JMP SHORT     NO, DO A SHORTENED TRANSFER
1092  00633 175671        STB EQT10,I   SAVE REMAINING CHAR. COUNT
1093  00634 066624R       LDB NMXFR     AND GET SET TO TRANSFER A LONG BURST
1094  00635 026644R       JMP BURST
1095*
1096  00636 140700  Z4XMR OCT 140700
1097  00637 165671  SHORT LDB EQT10,I   GET CHAR COUNT AGAIN
1098  00640 006007        INB,SZB,RSS   AND PUT OUT ALL BUT ONE OF THEM
1099  00641 026674R       JMP LAST1     OOPS! THIS -)IS(- THE LAST ONE
1100  00642 003400        CCA           SAVE -1 AS THE REMAINING COUNT
1101  00643 171671        STA EQT10,I
1102*
1103  00644 076732R BURST STB LFO       'LFO' IS COUNTER FOR # OF BYTES IN BURST
1104  00645 016656R       JSB ENQ?       CHECK FOR HANDSHAKING
1105                      IFN
1106                      JSB TRAP
1107                      DEF *-ORG
1108                      XIF
1109  00646 165670        LDB EQT09,I   BUFFER ADDR
1110  00647         WLOOP EQU *
1111  00647 105763        LBT           LOAD A BYTE FROM BUFFER
1112  00650 102600  OTA1  OTA 0
1113  00651 036732R       ISZ LFO       REQ'D # OF BYTES OUTPUT?
1114  00652 026647R       JMP WLOOP     NO, DO ANOTHER ONE.
1115  00653 175670        STB EQT09,I   STORE UPDATED BUFFER POSITION
1116*
1117  00654 016750R       JSB B.E.I     CO-ROUTINE AWAIT BUFFER EMPTY INTERRUPT
1118  00655 026625R       JMP WRIT      LOOP BACK FOR ANOTHER BURST
1119*
1120*
1121  00656 000000  ENQ?  NOP           ENQ-ACK HANDLER FOR HANDSHAKE N=2 OR 3
1122  00657 017251R       JSB HAND
1123  00660 000002        OCT 000002    SEE IF BIT 1 IS SET
1124  00661 002001        RSS           YES, XMIT REQUEST TO READ CHARACTER
1125  00662 126656R       JMP ENQ?,I    NO, RETURN
1126  00663 162012R       LDA EQX11,I   GET REQUEST CHAR (USUALLY ENQ)
1127  00664 017202R       JSB OTA       SEND TO FIFO
1128  00665 017150R       JSB DLAY      60 TIMES DELAY LOOP
1129  00666 017223R       JSB RCV       SET TO RECEIVE
1130                      IFN
1131                      JSB TRAP
```

```
1132                        DEF *-ORG
1133                        XIF
1134  00667 016750R         JSB B.E.I     CO-ROUTINE, RETURN TO RTE, WAIT FOR INTE
1135  00670 162000R BACK    LDA EQX01,I   "ACK!" RETURNS HERE (GET BAUD SETTING)
1136  00671 032636R         IOR Z4XMR     INCLUDE XMIT, CA, CD, & MASTER RESET BIT
1137  00672 017202R         JSB OTA       SET THE CARD
1138  00673 126656R         JMP ENQ?,I    RETURN TO CALLER
1139*
1140*
1141*
1142  00674 165670  LAST1 LDB EQT09,I    GET THE BUFFER POINTER BACK
1143  00675 016222R         JSB CNWD      TEST CONTROL BITS
1144  00676 002000  BIT10 OCT 002000     FOR TRANSPARENT MODE
1145  00677 026705R         JMP NBIN      NOT TRANSPARENT, HANDLE BACKARROW
1146  00700 105763          LBT
1147  00701 017202R         JSB OTA       PUT IT OUT
1148  00702 016750R         JSB B.E.I     WAIT FOR BUFFER EMPTY
1149  00703 026714R         JMP WCMPL     SIGNAL COMPLETED TRANSFER
1150*
1151  00704 000137  ARROW OCT 000137
1152*
1153  00705         NBIN  EQU *
1154  00705 105763          LBT           GET THE LAST CHARACTER
1155  00706 052704R         CPA ARROW     COMPARE TO BACKARROW/UNDERSCORE
1156  00707 026714R         JMP WCMPL     IT WAS, INHIBIT NORMAL END OF LINE
1157  00710 017202R         JSB OTA       WASN'T, PUT IT ON THE CARD
1158  00711 062165R CRLF    LDA C.R.      FOLLOW LAST CHAR. WITH
1159  00712 017202R         JSB OTA       CARRIAGE RETURN AND
1160  00713 016732R         JSB LFO       LINE FEED
1161*
1162  00714 000000  WCMPL NOP
1163                        IFN
1164                        JSB TRAP
1165                        DEF *-ORG
1166                        XIF
1167  00715 016656R         JSB ENQ?      HANDSHAKE AFTER FINAL CHARACTER (IF NEED
1168  00716 165667          LDB EQT08,I   GET ORIGINAL REQUEST LENGTH
1169  00717 006020          SSB           IF NEGATIVE,
1170  00720 007004          CMB,INB       MAKE POSITIVE
1171*
1172  00721         CMPLT EQU *         END OF TRANSFER
1173  00721 017223R         JSB RCV       SET TO KNOWN STATE = RECEIVE
1174  00722 017206R         JSB INTCN     SET UP INTERRUPTS
1175  00723 161667          LDA EQT08,I   GET REQUESTED LENGTH
1176  00724 002021          SSA,RSS       IF WORDS WERE REQUESTED,
1177  00725 005100          BRS           DIVIDE CHAR. BY 2
1178  00726 002400          CLA           A=0 FOR GOOD RETURN
1179  00727 171774          STA EQT15,I   TIME OUTS NOT EXPECTED NOW.
1180                        IFN
1181                        JSB TRAP
1182                        DEF *-ORG
1183                        XIF
1184  00730 126435R         JMP CX05,I    COMPLETION RETURN
1185*
1186  00165         C.R.  EQU CR
```

```
1187  00731 000012  L.F.  OCT 000012
1188*
1189  00732 000000  LFO   NOP           THIS SUBROUTINE OUTPUTS
1190  00733 062731R       LDA L.F.      A LINE FEED
1191  00734 017202R       JSB OTA
1192  00735 016750R       JSB B.E.I     WAITS FOR THE BUFFER TO EMPTY
1193  00736 162006R       LDA EQX07,I   CHECKS FOR A NEW LINE DELAY
1194  00737 002002        SZA
1195  00740 016742R       JSB T.O.I     AND WAITS FOR IT IF NEEDED
1196  00741 126732R       JMP LFO,I     THEN RETURNS TO CALLER
1197*
1198  00742         HALF  EQU *
1199  00742 000000  T.O.I NOP           TIME-OUT INTERRUPT CO-ROUTINE
1200  00743 171774        STA EQT15,I   SET DELAY VALUE
1201  00744 017114R       JSB SSTAT     SET BIT 0 IN STATUS WORD
1202  00745 000001  K1    OCT 000001    TO SAY WE EXPECT A TIME-OUT
1203  00746 066742R       LDB T.O.I     GET ADDRESS TO RETURN TO AFTER WAIT
1204  00747 026752R       JMP CONRT
1205*
1206  00750         MASK  EQU *
1207  00750 000000  B.E.I NOP           BUFFER EMPTY INTERRUPT ROUTINE.
1208  00751 066750R       LDB B.E.I     GET RETURN ADDRESS
1209  00752 175672  CONRT STB EQT11,I   SAVE CO-ROUTINE ADDRESS
1210  00753 036435R CEXIT ISZ CX05      BUMP RETURN TO CONTINUATION RETURN
1211  00754 017206R       JSB INTCN     CONFIGURE INTERRUPTS
1212  00755 126435R       JMP CX05,I    RETURN TO RTE TO AWAIT THE INTERRUPT
```

```
1214*
1215* THE HANDSHAKING MESSES UP THE ON-BOARD COUNTER, SO WE DONT USE IT.
1216* WE READ UNTIL THE FIFO IS EMPTY AND WE GET AN INVALID CHARACTER.
1217* THE COUNTER SIGNALS "HALF FULL" WITH N= 63 OR 64 EIGHT
1218* BIT WORDS AND "FULL" WITH N= 127 OR 128 WORDS.
1219*
1220  00756 002001  RSS   RSS
1221*
1222  00757 067021R INPUT LDB K177      PRESET FOR ASCII
1223  00760 016222R       JSB CNWD      SEE IF THE READ IS BINARY
1224  00761 002000        OCT 002000
1225  00762 026765R       JMP ASCII     NO, SET UP FOR ASCII READ
1226*
1227  00763 067020R       LDB K377      BINARY READ, SET MASK FOR ALL 8 BITS
1228  00764 062756R       LDA RSS
1229  00765 076750R ASCII STB MASK
1230  00766 022756R       XOR RSS       A=RSS FOR ASCII, NOP FOR BINARY
1231  00767 073003R       STA SWCH1
1232  00770 073036R       STA SWCH2
1233*
1234                      IFN
1235                      JSB TRAP
1236                      DEF *-ORG
1237                      XIF
1238  00771 165670        LDB EQT09,I   GET CURRENT ADDRESS IN BUFFER
1239  00772         RLOOP EQU *
1240  00772 102500  LIA2  LIA 00        GET A CHARACTER OFF THE CARD
1241  00773 002021        SSA,RSS       IF SIGN BIT NOT SET, NOT VALID CHAR.
1242  00774 027016R       JMP EMPTY     SO WE ARE DONE
1243  00775 172007R       STA EQX08,I   CHARACTER IS OK, SAVE FOR LATER USE
1244  00776 001206        OCT 1206  RAL,'ELA'  FUNNY INSTR. TO ROTATE LEFT AND
1245  00777 001300        RAR           COPY SIGN TO E.  THEN ROTATE BACK.
1246  01000 012750R       AND MASK      MASK OFF STATUS BITS.
1247  01001 002040        SEZ           IF E SET, IT IS A SPECIAL CHAR.
1248  01002 027027R       JMP SPCHR     YES, GO SEE WHICH ONE.
1249*
1250  01003 002001  SWCH1 RSS           WILL BE RSS FOR ASCII, NOP FOR BINARY
1251  01004 027012R       JMP NNULL
1252  01005 002002        SZA           IF CHARACTER IS A NULL,
1253  01006 052731R       CPA L.F.      OR A LINE FEED,
1254  01007 027015R       JMP EOCRD     DON'T TRANSFER TO USER BUFFER
1255  01010 052542R       CPA BIT3      IF CHARACTER IS A BACKSPACE (010)
1256  01011 027022R       JMP BAKUP     BACKUP THE BUFFER POINTER
1257*
1258  01012 156013R NNULL CPB EQX12,I   NORMAL OR BINARY, SEE IF ROOM IN BUFFER
1259  01013 027015R       JMP EOCRD     FOR MORE. NO?! - IGNORE UNTIL C.R.
1260  01014 105764        SBT           PUT CHARACTER IN USER'S BUFFER
1261  01015 026772R EOCRD JMP RLOOP     NOW SEE IF MORE ON THE CARD
1262  01016 175670  EMPTY STB EQT09,I   NO MORE ON CARD, UPDATE POINTER
1263                      IFN
1264                      JSB TRAP
1265                      DEF *-ORG
1266                      XIF
1267  01017 026753R       JMP CEXIT     AND TAKE CONTINUATION EXIT
1268*
```

```
1269*
1270  01020 000377  K377  OCT 000377
1271  01021 000177  K177  OCT 000177
1272*
1273  01022 156010R BAKUP CPB EQX09,I   SEE IF ALREADY AT START OF BUFFER
1274  01023 027015R       JMP EOCRD     YES, SO IGNORE
1275  01024 047026R       ADB M1        NO, BACK SPACE ONE CHAR POSITION
1276  01025 027015R       JMP EOCRD     GO SEE ABOUT NEXT CHAR.
1277*
1278  01026 177777  M1    OCT -1
1279*
1280*
1281  01027 013021R SPCHR AND K177      SO PARITY WON'T CONFUSE THE COMPARES
1282                      IFN
1283                      JSB TRAP
1284                      DEF *-ORG
1285                      XIF
1286  01030 052165R       CPA C.R.      FOR BINARY.  IS THIS A CARRIAGE RETURN?
1287  01031 027043R       JMP EOL       YES, DONE
1288  01032 052167R       CPA ACK       THIS MAY BE PERMIT TO XMIT FROM DEVICE
1289  01033 027077R       JMP ACK!      GO TO ACK
1290  01034 052170R       CPA DC2       THIS MAY BE DEVICE'S REQ. TO SEND
1291  01035 027107R       JMP DC2!      GO PROCESS DC2
1292  01036 002001  SWCH2 RSS           RSS FOR ASCII, NOP FOR BINARY
1293  01037 027012R       JMP NNULL     IF BINARY, EVEN RUBOUT GOES IN BUFFER
1294  01040 053021R       CPA K177      IS IT A RUB-OUT?
1295  01041 027065R       JMP RUBIT     YES, ISSUE \,CR,LF
1296  01042 027015R       JMP EOCRD     NEITHER?? SHOULDN'T GET HERE
1297*
1298*
1299  01043 161667  EOL   LDA EQT08,I   GET THE ORIGINAL REQUEST LENGTH
1300  01044 002020        SSA           WAS IT IN BYTES?
1301  01045 027051R       JMP NSTUF     YES, SO NO BLANK FILL
1302  01046 062463R       LDA BLANK     GET AN ASCII BLANK
1303  01047 004010        SLB           IS IT POINTING AT A RIGHT BYTE?
1304  01050 105764        SBT           STUFF BLANK IN USER'S BUFFER
1305  01051 162010R NSTUF LDA EQX09,I   GET THE STARTING BUFFER BYTE ADDR
1306  01052 003004        CMA,INA       MAKE NEGATIVE
1307  01053 044000        ADB AREG      END-START=#CHAR
1308  01054 162000R       LDA EQX01,I   GET WORD WITH HALF-DUPLEX
1309  01055 001200        RAL           BIT IN IT
1310  01056 002020        SSA           IF SET,
1311  01057 026721R       JMP CMPLT     DON'T ISSUE LINE FEED IN HALF DUPLEX
1312*
1313  01060 176010R       STB EQX09,I   SAVE THE TLOG INFO
1314                      IFN
1315                      JSB TRAP
1316                      DEF *-ORG
1317                      XIF
1318  01061 016203R       JSB MCLR      GO TO XMIT MODE, MAKE SURE WE CAN
1319  01062 016732R       JSB LFO       GET BUFFER EMPTY AFTER THE LINE FEED
1320  01063 166010R       LDB EQX09,I   GET B BACK
1321  01064 026721R       JMP CMPLT
```

```
1323* THE RUBOUT PROCESSOR IS RESPONSIBLE FOR RESETTING THE INPUT BUFFER
1324* POINTERS TO RE-ENTER A LINE.
1325* IT ISSUES A '\,CR,LF' TO THE USER'S TERMINAL.
1326*
1327  01065 016203R RUBIT JSB MCLR      TURN CARD AROUND
1328  01066 063076R       LDA BSLSH     GET A BACKSLASH
1329  01067 017202R       JSB OTA
1330  01070 062165R       LDA C.R.      CARRIAGE RETURN
1331  01071 017202R       JSB OTA
1332  01072 016732R       JSB LFO       LINEFEED AND DELAY
1333  01073 017130R       JSB TRIGR     PUT TERMINAL IN RECEIVE MODE AGAIN
1334  01074 166010R       LDB EQX09,I   GET THE ORIGINAL BUFFER ADDRESS
1335  01075 026772R       JMP RLOOP     AND START INPUT TRANSFER OVER AGAIN
1336*
1337  01076 000134  BSLSH OCT 000134    "\"
1338*
1339  01077 161672  ACK!  LDA EQT11,I   GET CO-ROUTINE RETURN ADDR
1340                      IFN
1341                      OTB 1         SAVE BREG @S
1342                      LDB .BACK     A=B IF GOOD ACK
1343                      JSB TRAP
1344                      DEF *-ORG
1345                      LIB 1         PULL B VALUE
1346                      XIF
1347  01100 105774        CBS M1 EQT11,I  CLEAR ADDRESS TO PREVENT RE-USE
      01101 001026R
      01102 101672
1348  01103 053106R       CPA .BACK     IS AN ENQ-ACK HANDSHAKE IN PROGRESS?
1349  01104 124000        JMP AREG,I    YES, CONTINUE
1350  01105 026753R       JMP CEXIT     NO, IGNORE THE WHOLE THING
1351*
1352  01106 000670R .BACK DEF BACK
1353*
1354  01107 161665  DC2!  LDA EQT06,I   IS THIS A READ OR WRITE?
1355                      IFN
1356                      JSB TRAP
1357                      DEF *-ORG
1358                      XIF
1359  01110 002011        SLA,RSS       IF BIT 0 IS CLEAR
1360  01111 026753R       JMP CEXIT     ITS A WRITE, IGNORE DC2
1361  01112 017130R       JSB TRIGR     ITS A READ. ISSUE PACE CH IF NEEDED
1362  01113 027015R       JMP EOCRD     DONT PUT IT IN USR BUFFER
```

```
1364  01114 000000  SSTAT NOP           SET BITS IN THE STATUS WORD
1365  01115 163114R       LDA SSTAT,I   GET BITS THE USER WANTS SET
1366  01116 131664        IOR EQT05,I   INCLUDE THEM IN STATUS WORD
1367  01117 171664        STA EQT05,I   PUT IT BACK
1368  01120 037114R       ISZ SSTAT     BUMP RETURN
1369                      IFN
1370                      JSB TRAP
1371                      DEF *-ORG
1372                      XIF
1373  01121 127114R       JMP SSTAT,I   AND RETURN
1374*
1375  01122 000000  CSTAT NOP           CLEAR BITS IN THE STATUS WORD
1376  01123 163122R       LDA CSTAT,I
1377                      IFN
1378                      JSB TRAP
1379                      DEF *-ORG
1380                      XIF
1381  01124 111664        AND EQT05,I   MASK OUT DESIRED BITS
1382  01125 171664        STA EQT05,I
1383  01126 037122R       ISZ CSTAT
1384  01127 127122R       JMP CSTAT,I
1385* THIS SUBROUTINE IS USED BY READS TO DETERMINE IF A PACE CHARACTER
1386* SHOULD BE SENT TO THE CARD.  IF IT SHOULD, IT DOES IT AND THEN
1387* SETS THE CARD INTO THE RECEIVE MODE.
1388*
1389  01130 000000  TRIGR NOP
1390                      IFN
1391                      JSB TRAP
1392                      DEF *-ORG
1393                      XIF
1394*********   START "DC1 VERSION"
1395  01131 017242R       JSB SUBCH     SEE IF THIS SUBCHANNEL
1396  01132 001000        OCT 001000    HAS BIT 9 SET
1397  01133 027140R       JMP PACE      YES, SO PACE IT
1398  01134 017251R       JSB HAND      NO, TRY HANDSHAKE
1399  01135 000001        OCT 000001    IS BIT 0 SET?
1400  01136 002001        RSS           YES, SO PACE IT
1401  01137 027146R       JMP NPACE     NO, DONT PACE
1402*
1403* MASTER RESET BEFORE "DLAY" INSURES THAT THE HARDWARE COUNTER
1404* IS OFF BY NO MORE THAN ONE COUNT AFTER FIFO TURN AROUND TRICK
1405* INPUT IS LIMITED TO 127 (NOT 128) CHARACTERS BEFORE COUNTER
1406* THINKS THE BUFFER IS FULL.
1407  01140 162000R PACE  LDA EQX01,I   GET BAUD SETTING
1408  01141 032636R       IOR Z4XMR     INCLUDE XMIT, CA, CD, & MASTER RESET BIT
1409  01142 017202R       JSB OTA       SET THE CARD
1410                      IFN           !!!!!DEBUG
1411                      OTA 1         SAVE THAT SETTING
1412                      XIF           !!!!!
1413  01143 162003R       LDA EQX04,I   GET THE TRIGGER CHARACTER
1414  01144 017202R       JSB OTA       SEND PACE CHAR INTO FIFO
1415                      IFN
1416                      JSB TRAP
1417                      DEF *-ORG
1418                      XIF
```

```
1419  01145 017150R        JSB DLAY      DO DELAY TRICK
1420*      HLT 10B        (((FOR DEBUG
1421                      IFN
1422                      JSB TRAP
1423                      DEF *-ORG
1424                      XIF
1425 01146 017223R NPACE JSB RCV       SET RECEIVE MODE AND ECHO BITS
1426  01147 127130R       JMP TRIGR,I
1427********* END DC1 VERSION
1428*
1429* "DELAY TRICK"----- WE DELAY LONG ENOUGH FOR THE CHARACTER TO COME OUT
1430                      IFN           !!!!!DEBUG
1431                      UNL
1432                      XIF           !!!!!
1433* OF THE FIFO INTO THE UART, THEN WE TURN THE CARD AROUND TO RECEIVE
1434* THE RESPONSE FROM THE TERMINAL BEFORE IT IS COMPLETELY OUT OF THE UART
1435* THIS IS THE ONLY WAY THAT WE CAN HOPE TO TURN THE CARD AROUND FAST
1436* ENOUGH TO KEEP FROM MISSING THE REPLY.  THE PROBLEM WITH THIS TRICK
1437* CAN BE STATED AS FOLLOWS:
1438************  ANYTIME YOU USE THIS TRICK, YOU MUST DO A MASTER CLEAR
1439* WARNING! *  BEFORE THE BUFFER EMPTY FLAG CAN BE SEEN!  WE HAVE PUT
1440************  A CHAR IN THE FIFO, WHICH CAUSES THE COUNTER TO COUNT UP
1441************  BY ONE, BUT THE CARD NEVER SEES IT COME OUT, BECAUSE THE
1442* WARNING! *  FIFO HAS BEEN TURNED AROUND!  THERE MUST BE A CALL TO MCLR
1443************  BETWEEN THIS CALL AND ANY SUCCEEDING WRITES. (ANT)
1444************  ALSO, IN A LONG SERIES OF READS, THE COUNTER ACCUMULATES
1445* WARNING! *  COUNTS UNTIL IT THINKS IT IS FULL AND THAT, TOO, WILL
1446************  CAUSE THE DRIVER TO HANG. AIN'T TECHNOLOGY GRAND! (RCB)
1447*
1448* AS OF NOV 81, THE PROBLEM IS RESOLVED AS OUTLINED IN THE INITIAL
1449* COMMENTS TO THIS DRIVER; HOWEVER, THERE IS ANOTHER PROBLEM IF THE
1450* DRIVER IS USED IN A PRIVELEGED SYSTEM:
1451* A PRIVELEGED DRIVER CAN SUSPEND THIS ONE AND CAUSE THE APPROXIMATELY
1452* 96 MICROSECOND DELAY TO BE MUCH LONGER. DATA MAY BE LOST THIS WAY.
1453*
1454                      IFN           !!!!!DEBUG
1455                      LST
1456                      XIF           !!!!!
1457*
1458  01150 000000  DLAY  NOP           "60 TIMES DELAY LOOP"
1459  01151 063155R       LDA M60       WAIT 60 TIMES
1460  01152 002006        INA,SZA       THROUGH 1.645 USEC LOOP (ON MX-E
1461  01153 027152R       JMP *-1
1462  01154 127150R       JMP DLAY,I    WITH HIGH SPEED MEM.) 98.7USEC TOTAL
1463*
1464  01155 177704  M60   DEC -60



1466  01156 000000  LINCK NOP           RS232 LINE CHECK ROUTINE
1467  01157 162007R       LDA EQX08,I   GET CARD STATUS
1468  01160 013167R       AND RS232     MASK TO RS232 STATUS LINES
1469                      IFN
1470                      OTB 1         SAVE B IN S REG
1471                      LDB RS232     !COMPARE B TO A. IF EQUAL
```

```
1472                    JSB TRAP        !THEN LINE STATUS IS OK
1473                    DEF *-ORG
1474                    LIB 1           RESTORE BREG
1475                    XIF
1476  01161 053167R     CPA RS232       SEE IF ALL WERE SET
1477  01162 037156R     ISZ LINCK       YES, MAKE SURE WE DO A SKIP RETURN
1478*
1479  01163 172011R     STA EQX10,I     SAVE CURRENT STATUS
1480  01164 032220R     IOR Z2          MAKE A COMMAND TO SET
1481  01165 017202R     JSB OTA         THE DEVICE REFERENCE REGISTER
1482  01166 127156R     JMP LINCK,I
1483*
1484                    IFN             !!!!!DEBUG
1485                    UNL
1486                    XIF             !!!!!
1487* WORD 'RS232' IS USED TO DETERMINE WHICH INPUT LINES TO THE CARD
1488* MUST BE SET TO INDICATE THE TERMINAL IS OK AND Z1 DEFINES WHICH
1489* LINES ARE ENABLED TO CAUSE INTERRUPTS.  THEY SHOULD AGREE!!
1490*  BB BBB    BITS 4-0 ARE THE SENSE LINES
1491*       ^-0- SCF (SECONDARY LINE SIGNAL DETECTOR)
1492*      ^--1- CF  (RECEIVED LINE SIGNAL DETECTOR)
1493*     ^---2- CE  (RING INDICATOR)
1494*   ^-----3- CC  (DATA SET READY)
1495*  ^------4- CB  (CLEAR TO SEND)
1496*
1497* THERE ARE STRAPPING OPTIONS TO THE CARD THAT CAN MAKE OTHER
1498* SIGNALS AVAILABLE FOR CHECKING.  SEE HP MANUAL #12966-90001.
1499*
1500                    IFN             !!!!!DEBUG
1501                    LST
1502                    XIF             !!!!!
1503*
1504  01167 000032  RS232 OCT 000032
1505  01170 010032  Z1    OCT 010032



1507* THIS ROUTINE READS THE CARD STATUS.  THE FORMAT IS:
1508*
1509*  RS232 MODEM CABLE (OPT.002) PIN NUMBERS    5   6   22  8   12
1510*---:---.---.---:---.---.---:---.---.---:---.---.---:---.---.---:
1511*DEV:SPC NOT  R :    NOT BFF:BFF BFF    :OVR        :        SBB:
1512*INT:CHR USD SPR:TST USD HLF:FUL MTY BRK:PAR  CB  CC: CE CF  SCF:
1513*---:---.---.---:---.---.---:---.---.---:---.---.---:---.---.---:
1514*
1515  01171 000000  STATS NOP
1516  01172 106700  CLC1  CLC 00       CLEAR CONTROL TO TELL CARD WE
1517  01173 017176R       JSB LIA       WANT STATUS, NOT DATA
1518  01174 102700  STC1  STC 00       SET BACK TO NORMAL MODE
1519                    IFN             !!!!!DEBUG
1520                    JSB TRAP        ! STATUS IS IN A REG
1521                    DEF *-ORG       ! B,S UNDEFINED
1522                    XIF
1523  01175 127171R     JMP STATS,I
1524*
```

```
1525*



1527* THIS ROUTINE READS WORDS FROM THE CARD.  IF CONTROL IS CLEAR, THE WORD
1528* READ IS A STATUS WORD AS SHOWN ABOVE.  IF CONTROL IS SET, THE WORD IS
1529* A DATA WORD.  THE FORMAT OF THE DATA WORD IS:
1530*---:---.---.---:---.---.---:---.---.---:---.---.---:---.---.---:
1531*VLD:SPC.      CHAR COUNT        .             DATA              :
1532*CHR:CHR.      IN BUFFER         .             INPUT             :
1533*---:---.---.---:---.---.---:---.---.---:---.---.---:---.---.---:
1534*
1535  01176 000000  LIA   NOP
1536  01177 102500  LIA1  LIA 00         GET DATA OR STATUS FROM CARD
1537  01200 172007R       STA EQX08,I    SAVE FOR OTHER ROUTINES
1538  01201 127176R       JMP LIA,I



1540*  THIS ROUTINE DOES ALL OUTPUT TO THE CARD.  IT SAVES THE WORD
1541*  THAT IS OUTPUT FOR DEBUGGING.
1542  01202 000000  OTA   NOP
1543  01203 102600  OTA2  OTA 00         PUT OUT THE WORD!
1544  01204 172002R       STA EQX03,I    SAVE FOR DEBUGGING
1545  01205 127202R       JMP OTA,I



1547* THIS ROUTINE IS USED TO CLEAR ALL INTERRUPTS ON THE CARD
1548* AND ENABLE THE CARD TO INTERRUPT.
1549* IT SHOULD BE CALLED BEFORE ALL EXITS FROM THE DRIVER.
1550*
1551  01206 000000  INTCN NOP            CONFIGURE CARD FOR INTERRUPTS
1552  01207 063213R       LDA Z5         GET CONTROL WORD TO
1553  01210 017202R       JSB OTA        CLEAR ALL FLAGS
1554                      IFN            !!!!!DEBUG
1555                      JSB TRAP
1556                      DEF *-ORG
1557                      JSB STATS      !!!!!FOR DEBUG
1558                      XIF
1559  01211 103700  STCC1 STC 00,C       ENABLE CARD TO INTERRUPT
1560  01212 127206R       JMP INTCN,I    ALL READY TO DO IT!
1561*
1562  01213 050077  Z5    OCT 050077     CLEAR ALL FLAGS



1564* THIS ROUTINE IS USED TO PUT THE INTERFACE IN THE TRANSMIT MODE,
1565* AT THE PROPER BAUD.
1566*
1567  01214 000000  XMIT  NOP
1568  01215 162000R       LDA EQX01,I    GET THE BAUD SETTING
1569  01216 033221R       IOR Z4X        CONTROL WORD (XMIT, CA & CD ON)
1570  01217 017202R       JSB OTA        SET THE CARD
1571  01220 127214R       JMP XMIT,I
```

```
1572*
1573  01221 040700  Z4X   OCT 040700
1574  01222 040300  Z4R   OCT 040300



1576*  THIS SUBROUTINE PUTS THE INTERFACE INTO RECEIVE MODE AT THE
1577*  PROPER BAUD.
1578*
1579  01223 000000  RCV   NOP
1580  01224 162000R       LDA EQX01,I   GET THE BAUD SETTING FROM EQX01
1581  01225 033222R       IOR Z4R       INCLUDE RCV, CA,CD ON, BITS
1582  01226 017202R       JSB OTA       PUT OUT TO CARD
1583*
1584  01227 162000R       LDA EQX01,I   GET WORD WITH HALF-DUPLEX BIT
1585  01230 001200        RAL           MOVE IT TO SIGN BIT
1586  01231 002020        SSA           IF IT IS SET,
1587  01232 027235R       JMP NECHO     NEVER ECHO-PLEX
1588*
1589  01233 016222R       JSB CNWD      CHECK THE CONTROL WORD
1590  01234 000400  BIT8  OCT 000400    TO SEE IF THE ECHO BIT IS ON
1591  01235 002401  NECHO CLA,RSS       NO, SKIP
1592  01236 062620R       LDA BIT4      YES, INCLUDE THE ECHO BIT IN
1593  01237 132005R       IOR EQX06,I   THE CONTROL WORD FOR CHAR. SIZE, ETC.
1594  01240 017202R       JSB OTA       AND SET THE CARD
1595  01241 127223R       JMP RCV,I



1597*  THIS ROUTINE CHECKS FOR A BIT SET IN EQT04.
1598*     JSB SUBCH
1599*     OCT XXXX      MASK TO USE
1600*     ( RETURN IF ANY BITS SET )
1601*     ( RETURN IF NO BITS SET  )
1602*
1603  01242 000000  SUBCH NOP           CHECK EQT04 BITS
1604                      IFN
1605                      JSB TRAP
1606                      DEF *-ORG
1607                      XIF
1608  01243 163242R       LDA SUBCH,I   GET USERS MASK
1609  01244 037242R       ISZ SUBCH
1610  01245 111663        AND EQT04,I   AND WITH EQT04
1611  01246 002003        SZA,RSS       IF BITS NOT SET,
1612  01247 037242R       ISZ SUBCH     BUMP RETURN
1613  01250 127242R       JMP SUBCH,I
1614*
1615*
1616  01251 000000  HAND  NOP           CHECK EQX13 BITS
1617  01252 163251R       LDA HAND,I    GET USERS MASK
1618  01253 037251R       ISZ HAND      ADJUST RETURN ADDRESS
1619  01254 112014R       AND EQX13,I   AND WITH HANDSHAKING MASK
1620  01255 002003        SZA,RSS       IF NO RESULTING BITS ARE SET
1621  01256 037251R       ISZ HAND      BUMP RETURN ADDR
1622                      IFN
```

```
1623                    JSB TRAP
1624                    DEF *-ORG
1625                    XIF
1626  01257 127251R     JMP HAND,I    TAKE COMPUTED RETURN
1627*
1628*
```

```
1630  01675          SYSTY EQU 1675B
1631  01734          OPATN EQU 1734B
1632*
1633  01260 065660   BREAK LDB EQT01     SEE IF OUR EQT ADDRESS
1634                      IFN
1635                      LDA EQX02,I    GET THE ID SEG ADDR FROM EQT EXTENSION
1636                      JSB TRAP
1637                      DEF *-ORG
1638                      XIF
1639  01261 055675         CPB SYSTY      IS THE SAME AS THE SYSTEM CONSOLE.
1640  01262 027300R        JMP SYSTM      YES
1641*
1642  01263 162001R        LDA EQX02,I    GET THE ID SEG ADDR FROM EQT EXTENSION
1643  01264 073275R        STA PRAM1      IT WILL BE -1 IF NO ID TO SCHEDULE
1644  01265 165664         LDB EQT05,I    ALSO GET THE ENABLE FLAG IN EQT05 BIT 1
1645  01266 005310         RBR,SLB        BOTH ENABLE BIT SET,
1646  01267 002020         SSA            AND ID ADDRESS OK?
1647  01270 026530R        JMP BRKRT      NO, EXIT THE WHOLE THING
1648                      IFN
1649                      JSB TRAP
1650                      DEF *-ORG
1651                      XIF
1652*
1653  01271 065663         LDB EQT04      YES, SO DO A SCHEDULE CALL
1654  01272 077276R        STB PRAM2      PASSING EQT04 ADDRESS FOR EQTLU
1655*
1656* THIS SECTION COULD BE ENHANCED TO SCHEDULE PROGRAMS BY NAME,
1657* AND THE NAMES COULD BE PASSED TO THE DRIVER IN A CONTROL CALL,
1658* TO ALLOW EASIER IMPLEMENTATION OF SYSTEMS THAT MAKE RTE "INVISIBLE"
1659* TO THE USER.
1660*
1661  01273 016002X        JSB $LIST      GO SCHEDULE THE PROGRAM
1662*     JSB LINK2,I
1663  01274 000601         OCT 000601
1664  01275 000000   PRAM1 NOP            * ALSO USED AS TEMPORARY STORAGE ELSEWHE
1665  01276 000000   PRAM2 NOP
1666*     HLT 23B        <<< FOR DEBUGGING
1667  01277 026530R         JMP BRKRT
1668*LINK2 DEF $LIST     <<< FOR DEBUG
1669*
1670*
1671  01300 035734   SYSTM ISZ OPATN      WHEN WE SET THIS FLAG,
1672  01301 026530R         JMP BRKRT      THE SYSTEM ISSUES A '*' PROMPT
1673                      END IX05
**  NO ERRORS #TOTAL **RTE ASMB 92067-16011**
```