

Central Data 2650 Newsletter

Here we are with another issue of the newsletter--sent out very close to schedule for a change! I have been very impressed by the type of programs that everyone has sent me, and hope that you keep it up.

A couple of programs of particular interest are the Morse Code program written by Mike Durham and the Selectric Driver routine written by Dennis Haynes. From the calls that I get, I'm sure that these programs will interest many of our readers.

Also, a note should be made that all of our software updates will be done through the newsletter from now on. This proves to be much easier for us, and since everyone who is still using their system to any extent receives the newsletter, this should be no problem to any of our customers.

I hope that you enjoy this issue of our 2650 newsletter, and again I encourage you to send in your comments, suggestions, and programs.

LETTERS

Dear Jeff:

While reading your newsletter, I came across the letter from Mike Kelley asking for black on white letters. I came up with a simple modification to make this a switch-selectable option: Cut the trace from pin 13 of IC60 where it runs nearest IC63. The signal from IC60 goes to pin 5 of IC63 and to one side of a SPDT switch. The switch common terminal is connected to the trace that goes on to the video transistors. Using black on white gives me slightly skewed display (sometimes), but can be a help under bright lighting conditions.

I have also enclosed a rather faint listing of some assembler improvements you might be interested in. It has been assembled to occupy the remaining space in the block occupied by your Version 1.2 patches. 3327 to 338C produces a symbol table printout. 338D to 33E8 print the number of errors detected, but will only give a rough approximation after 15 errors because of the simplistic method used. (Oh, I almost forgot that you must use "T" command to print the symbols, but the program uses the user's print subroutine.) Patches to the assembler are at 21C5-33 27, at 21C9-33 8D, and at 321B-1F 33 AE C0.

Yours truly,

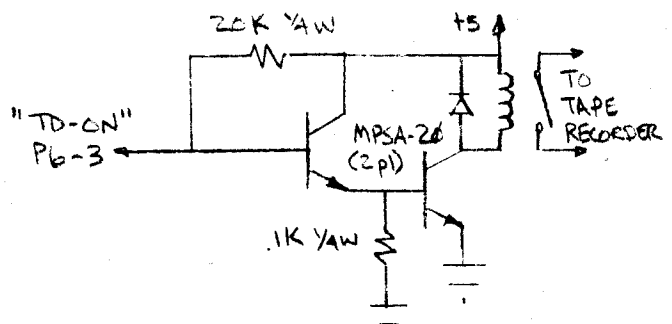
Gordon Brandly

Dear Gordon:

Thanks for the mod to get black on white characters. I am sorry to say that your listing was too faint to print, but we will be glad to run it next issue if you could send us a darker copy. Thanks again!

Gentlemen:

I used a MPSA-20 transistor and a small 4½ volt relay in the cassette control circuit, and could not get it to work properly. The setup as you have in the manual doesn't work because it loads the voltage down on pin 3 of plug 6. I had to modify the circuit as below and it works fine.



Once I got this far I found that dumping the data on tape was no problem. However, after following your instructions very carefully, I found that I could not load the data back into the machine. I've compared the schematic of the tape interface in your manual with that in Radio Electronics Magazine and there is considerable difference. Which is correct?

I would appreciate any assistance that you could give on this. Also, does your company produce an 8K memory for the unit?

Sincerely yours,

Kenneth E. Walters
Box 54
Red Ash, VA 24640

Dear Kenneth:

Thanks for the suggestion concerning the cassette tape on/off circuit. Answering your questions, our Computer System Manual is much more accurate than the Radio-Electronics articles. Also, we do not sell an 8K memory board although we do offer a 16K or 32K board for the 2650 at a very low price. ED.

Gentlemen:

Below is a random number generator program for the 2650 computer. The original program was for use with a 6800 microprocessor. I merely transposed the 6800 instructions to 2650 instructions. It works very well on my machine. It may be of interest to other 2650 users.

CPSL	18	clr WC & RS
LODA,R2	1528	
LODZ,R2		
SUBI,R2	1	
ADDZ,R2		
SUBI,R2	1	
SUBI,R2	1	
ADDZ,R2		
ADDI,RO	1	
STRA,RO	1528	save current no.
RETC,UN		return

Sincerely,

Ken Walters

Dear Ken:

Thanks for the random-number generating program. It really comes in handy for a lot of programs! If anyone has come up with one for 16 bit numbers, please send it in.

Dear Jeff:

I am enclosing a copy of the ASCII to IBM selectric routine I told you about. If you or any of the other CDC system owners want a copy of the source code I'll be happy to send it if a blank disk or tape is provided.

The other routine 'DUMP' is a quick and dirty program to print characters from memory. To run it just load the file into memory using the read command (disk systems only) beginning at H4400. Then enter "RUN DUMP". This routine makes it possible to write letters or other text using the editor and then print them using this routine.

Regards,

Dennis Haynes

Dennis:

Thanks for the great programs! I have received numerous calls asking for things just like that. Also, we appreciate your offer for sending source code out. That will help a lot of people.

Subscription Information

The Central Data 2650 Newsletter is published every 8 weeks by Central Data Corporation. Any comments should be sent to Central Data Corp.; PO Box 2530 Station A; Champaign, IL 61820.

Subscription Rate: \$4.00 per year.

Editor: Jeff Roloff.

Symbol Table Printer by Raymond Rouillard

```

0021      FRMT
0022  *
0023  * * * * *
0024  *
0025  *      SYMBOL TABLE ROUTINE TO USE THE
0026  *      T  COMMAND OF THE CENTRAL DATA
0027  *      EDITOR.ASSEMBLER VERSION 1.2
0028  *      BY RAYMOND N. ROUILLARD
0029  *      5-1-78 REVISED 5-23-78
0030  *
0031  * * * * *
0032  *
0033  *
0034  *
0035  *
0036  *
0037  *
0038  *
0039  *
0040  *
0041  *
0042  *
0043  *
0044  *
0045  *
0046  *
0047  *
0048  *
0049  *
0050  *
0051  *
0052  *
0053  *
0054  *
0055  *
0056  *
0057  *
0058  *
0059  *
0060  *
0061  *
0062  *
0063  *
0064  *
0065  *
0066  *
0067  *
0068  *
0069  *
0070  *
0071  *
0072  *
0073  *
0074  *
0075  *
0076  *
0077  *
0078  *
0079  *
0080  *
0081  *
0082  *
0083  *
0084  *
0085  *
0086  *
0087  *
0088  *
0089  *
0090  *
0091  *
0092  *
0093  *
0094  *
0095  *
0096  *
0097  *
0098  *
0099  *
0100  *
0101  *
0102  *
0103  *
0104  *
0105  *
0106  *
0107  *
0108  *
0109  *
0110  *
0111  *
0112  *
0113  *
0114  *
0115  *
0116  *
0117  *
0118  *
0119  *
0120  *
0121  *
0122  *

```

```

0057      STRZ,3
0058      LGDA,1      *PNT2,I
0059      STRA,1      *PNT1,I
0060      LOD7,3
0061      STRA,1      *PNT2,I
0062      TMI ,1      7
0063      BCFR,0      SWITCH
0064      STRR,1      TEMP
0065      LODR,1      HOLD
0066      ADDI,1      00
0067      BCTA,UN     CONT
0068      DATA      02
0069      DATA      00
0070      COMF      LOPR,0
0071      RCFA,0      START
0072  *
0073      FORZ,0      WRITE ROUTINE
0074      STRR,0      HOLD
0075      ZBSR      LFCR
0076      LODI,1      FF
0077      LCLA,2      TABLE
0078      SUBI,2      01
0079      STRR,2      TEMP
0080      BEG      COMI,1
0081      BCFR,0      ACN
0082      LODR,2      TEMP
0083      ADDI,2      01
0084      STRR,2      TEMP
0085      ACN      LCDA,1
0086      COMI,0      *TEMP,+
0087      BCTR,0      FF
0088      TMI ,1      00
0089      BCTR,0      NEXT
0090      STRZ,3      STRZ,3
0091      BSTA,UN     WCHR
0092      BCTR,UN     ACN
0093      NEXT      STRR,1
0094      STRZ,2      TET
0095      BSTR,UN     SUBR
0096      BSTA,UN     HIOT
0097      LODR,1      TET
0098      LOCA,1      *TEMP,+
0099      STRR,1      TET
0100      STRZ,2      STRZ,2
0101      BSTA,UN     HIOT
0102      BSTR,UN     SUBR
0103      LODR,1      TET
0104      TMI ,1      18
0105      BCFR,0      BEG
0106      LODR,0      *CURS
0107      TMI ,0      07
0108      BSTA,0      PAUSE
0109      ZBSB      LFCR
0110      BCTA,UN     BEG
0111      TET      DATA
0112      SUBR      BSTA,UN
0113      BSTA,UN     WBLI
0114      BSTA,UN     WBLI
0115      BSTA,UN     WBLI
0116      RETC,UN     PAUSE
0117      BSTA,UN     RETURN
0118      BCTA,UN     00
0119      FILL      DATA
0120      CURS      ACON
0121      *      17FF
0122      TAF      ORG      21C5
0123      ACON      START

```

COMMAND TABLE CHANGE

Number Guessing Game by Jerry Johnson

(AS) THIS PROGRAM WAS WRITTEN FOR THE CENTRAL-DATA 2450
(AS) COMPUTER BY JERRY J JOHNSON ON 25 FEB 78. IT IS
(AS) WRITTEN IN 8K BASIC.

(AS)
(AS) (AS) MEANS ASTERIX
(AS) (PD) MEANS POUNDS
(AS) (EQ) MEANS EQUAL
(AS) (PL) MEANS PLUS
(AS) (LT) MEANS LESS THAN
(AS) (GT) MEANS GREATER THAN
(AS)

PRINT ' IN THIS GAME I WILL GENERATE A NUMBER IN THE'
PRINT 'RANGE OF 1 TO 1000. I WILL ALLOW YOU FROM'
PRINT '5 TO 15 GUESSES TO DETERMINE THE NUMBER.'
PRINT 'I WILL TELL YOU IF YOU ARE TOO HIGH OR TOO LOW'
PRINT 'AFTER EACH ENTRY PRESS CARRIAGE RETURN.'
PRINT 'TO TERMINATE AT ANY TIME, PRESS THE ESCAPE KEY.'

PRINT
W(EQ)0
F(EQ)0
INPUT 'WHAT IS YOUR NAME ? ' NS
Y(EQ)0
20 ERASE
30 A(EQ)INT(RND(999))
50 B(EQ)INT(RND(15))
IF B(LT)5 GOTO 50
55 IF B(EQ)0 GOTO 540
PRINT 'YOU HAVE '(PD)B' GUESSES REMAINING'
INPUT 'WHAT IS YOUR GUESS ? ' G
Y(EQ)Y(PL)1
ERASE
B(EQ)B-1
IF A(LT)G GOTO 200
IF A(GT)G GOTO 300
W(EQ)W(PL)1
PRINT 'CORRECT 'NS' IT TOOK YOU '(PD)Y' TRYS !'
PRINT '///// WON '(PD)W' /////-///// LOST '(PD)F' /////'
GOTO 510
200 PRINT 'YOUR GUESS OF '(PD)G' IS TOO BIG'
GOTO 55
300 PRINT 'YOUR GUESS OF '(PD)G' IS TOO SMALL'
GOTO 55
510 INPUT 'PLAY AGAIN ? (Y/N) ' X1
Y(EQ)0
IF X1(EQ)'Y' GOTO 20
IF X1(EQ)'N' GOTO 600
GOTO 510
540 F(EQ)F(PL)1
550 PRINT 'SORRY, YOU LOOSE 'NS' THE NUMBER WAS '(PD)A'
PRINT '///// WON '(PD)W' /////-///// LOST '(PD)F' /////'
GOTO 510
600 PRINT 'T H A N K S F O R T H E G A M E !'
STOP

Lots of Programs by Mike Herbach

EXEC CHANGES

THERE HAS BEEN TALK OF PUTTING 1702 OR 2708 PROMS IN PLACE OF THE CENTRAL DATA EXEC IN ORDER TO MAKE SOME IMPROVEMENTS AT THE COST OF AN EXTRA POWER SUPPLY. FOR ANYONE INTERESTED, HERE ARE THE CHANGES IN MY SYSTEM.

1) WHEN YOU TURN ON YOUR SYSTEM OR PUSH THE POWER RESET, THE FIRST FEW COMMANDS SET THE PROGRAM STATUS WORDS. HOWEVER, THEY DO NOT SET THE MEMORY LOCATIONS AT 17F2 AND 17F3. THESE ARE USED FOR STORING AND EXAMINING THE PSL AND PSU WHEN YOU USE THE BREAKPOINT ROUTINE. THEY ARE STUFFED BACK INTO THE PSL AND PSU WHEN YOU USE THE "E" COMMAND TO RUN A PROGRAM. THIS MEANS THAT THE FIRST TIME YOU EXECUTE A PROGRAM THE INITIAL RANDOM VALUES AT THESE LOCATIONS BECOME YOUR PROGRAM STATUS WORDS. TO FIX THIS, CHANGE PROM LOCATIONS AT 0000 TO 04 42, 92, 93, CC 17 F2, CC 17 F3, 05 11, CD 17 FE, 18 02.

2) MEMORY LOCATION 1500 TO 150F SHOULD NOT BE USED BECAUSE THEY ARE SOMETIMES CHANGED TO 20'S. THIS HAPPENED WHENEVER THE 80TH COLUMN IS USED AND THE MACHINE DOES AN AUTOMATIC LINEFEED/CARRIAGE RETURN. AT THIS POINT, THE LFCR ROUTINE ERASES THE CURSER MARK IT THINKS IS AT THE OLD LOCATION. THE SPACE IT WRITES IS THE 20 THAT MESSES UP THOSE LOCATIONS AT 1500 SINCE THEY ARE THE FIRST LOCATIONS AT THE END OF EACH LINE ON THE SCREEN. TO FIX THIS MAKE SURE YOU NEVER USE 18 80TH COLUMN OR CHANGE EXECUTIVE PROM LOCATION 0387 FROM 24 TO 29. THIS IS THE 439TH DECIMAL LOCATION OF THE SECOND EXEC PROM. ALSO, NEVER DO A LFCR WHILE THE CURSER IS ON THE 80TH COLUMN (CURSER LOCATIONS 14F0-14FF).

3) THE REQUIREMENT TO HIT THE "C" KEY BEFORE ANY RAM MEMORY CHANGE CAN BE A BIT MUCH WHEN ENTERING LARGE PROGRAMS OR RAM PROGRAM CHANGES. TO AVOID THIS MAKE THE FOLLOWING CHANGES AT EXEC LOCATION #10DD (DECIMAL 221 IN THE FIRST PROM): E7 20 18 05, 3B 56. AFTER THIS WHEN USING THE "A" COMMAND YOU CAN CHANGE THE VALUE DISPLAYED BY TYPING ONLY THE NEW VALUE AND THE NEXT VALUE IS AUTOMATICALLY DISPLAYED. USE THE SPACE BAR TO INCREMENT LOCATIONS WITHOUT CHANGES. BEWARE: ANY NON HEX KEY APPEARS AS A 00.

MIKE HERBACH

Programs

Screen Printer by Ray Krygier

```

0002 0000      *
0003 0000      * ***** COPY *****
0004 0000      *
0005 0000      * THIS PROGRAM WILL COPY THE SCREEN TO A HARDCOPY PRINTER REMOVING
0006 0000      * RIGHT BLANKS .
0007 0000      *
0008 0000      * WRITTEN BY ;
0009 0000      * RAY KRYGIER
0010 0000      * 25353 CROWN POINT COURT
0011 0000      * FARMINGTON HILLS, MICHIGAN
0012 0000      *          48018
0013 0000      *
0014 1500      ORG      1500
0015 1500      RO      EQU      0
0016 1500      R1      EQU      1
0017 1500      UN      EQU      3
0018 1500      GT      EQU      01
0019 1500      EQ      EQU      00
0020 1500      *
0021 1500 3C 95 58 WRT  BSTA  *LNCR  INITIALIZE ROUTINE
0022 1503 04 10      LODI,RO  10
0023 1505 CC 15 56      STRA,RO  SCRNI
0024 1508 20      EORZ,RO
0025 1509 CC 15 57      STRA,RO  SCRNI
0026 150C CC 15 5C      STRA,RO  CTR
0027 150F      *
0028 150F 3F 15 5D WRTA  BSTA,UN  SCAN  GET NUMBER OF PRINTABLE CHAR.
0029 1512 0C 95 56 WRT1  LODA,RO  *SCRNI  UNFOLD CHAR.
0030 1515 F4 20      TMI ,RO  20
0031 1517 18 02      BCTR,EQ  WRT2
0032 1519 64 40      IORI,RO  40
0033 151B 3F 95 5A WRT2  BSTA,UN  *WRITER  WRITE A CHAR.
0034 151E 0C 15 57      LODA,RO  SCRNI  ADD 10 TO ADDRESS
0035 1521 84 10      ADDI,RO  10
0036 1523 CC 15 57      STRA,RO  SCRNI
0037 1526 77 08      PPSL      08
0038 1528 0C 15 56      LODA,RO  SCRNI
0039 152B 84 00      ADDI,RO  00
0040 152D CC 15 56      STRA,RO  SCRNI
0041 1530 75 08      CPSL      08
0042 1532 0D 15 5C      LODA,R1  CTR      SUBTRACT 1 FROM CTR.
0043 1535 A5 01      SUBI,R1  01
0044 1537 CD 15 5C      STRA,R1  CTR
0045 153A 1D 15 12      BCTA,GT  WRT1      DO NEXT CHAR.
0046 153D 0C 15 57      LODA,RO  SCRNI  LAST LINE ?
0047 1540 F4 0F      TMI ,RO  *
0048 1542 14      RETC,EQ  10      YES RETURN
0049 1543 04 10      LODI,RO  10      BUMP TO NEXT LINE
0050 1545 CC 15 56      STRA,RO  SCRNI
0051 1548 8C 15 57      LODA,RO  SCRNI
0052 1548 84 01      ADDI,RO  01
0053 154D CC 15 57      STRA,RO  SCRNI
0054 1550 3F 95 58      BSTA,UN  *LNCR
0055 1553 1F 15 0F      BCTA,UN  WRTA      DO NEXT LINE
0056 155C 00      SCRNI  DATA  00
0057 1557 00      SCRNI  DATA  00

0058 1558 41 47      LNCR  ACON      4147  ADDRESS OF LINEFEED & CARRIER ROUTINE
0059 155A 40 00      WRITER ACON  4000  ADDRESS OF WRITER ROUTINE
0060 155C 00      CTR  DATA  00
0061 155D      *
0062 155D 05 51      SCAN  LODI,R1  51  SCAN LINE BACKWARD
0063 155F 04 15      LODI,RO  15  SET TO END OF LINE
0064 1561 CC 15 56      STRA,RO  SCRNI
0065 1564 0C 15 57      LODA,RO  SCRNI
0066 1567 44 0F      ANDI,RO  0F  RESET LINE POSITION
0067 1569 CC 15 57      STRA,RO  SCRNI *
0068 156C A5 01      SUBI,R1  01
0069 156E 0C 15 57      LODA,RO  SCRNI
0070 1571 A4 10      SUBI,RO  10
0071 1573 CC 15 57      STRA,RO  SCRNI
0072 1576 77 08      PPSL      08
0073 1578 0C 15 56      LODA,RO  SCRNI
0074 157A A4 00      SUBI,RO  00
0075 157D CC 15 56      STRA,RO  SCRNI
0076 1580 75 08      CPSL      08
0077 1582 E5 01      COMI,R1  01
0078 1584 18 08      BCTR,EQ  *SCAN2
0079 1586 0C 95 56      LODA,RO  *SCRNI
0080 1589 E4 20      COMI,RO  20
0081 158B 1C 15 6C      BCTA,EQ  *SCRNI
0082 158E CD 15 5C SCAN2  STRA,R1  CTR      SAVE NUMBER OF PRINTABLE CHAR.
0083 1591 04 10      LODI,RO  10
0084 1593 CC 15 56      STRA,RO  SCRNI  RESET TO START OF LINE
0085 1596 0C 15 57      LODA,RO  SCRNI
0086 1599 44 0F      ANDI,RO  0F  RESET LINE POSITION
0087 159B CC 15 57      STRA,RO  SCRNI
0088 159E 17      RETC,UN
0089 159F      *
0090 159F      * THIS IS PRINTED ON A SELECTIC PRINTER USING CENTRAL DATA'S BIT
0091 159F      * SELECTABLE I/O PORT .
0092 159F      *

```

Hamurabi by Bruce Blakeslee

HERE IS THE PROGRAM:

* HAMURABI - GOVERN THE KINGDOM OF SUMERIA
* MODIFIED BY BRUCE BLAKESLEE FROM "WHAT TO DO AFTER YOU HIT RETURN."
* FOR CENTRAL DATA SYSTEM AND BK BASIC.
* VARIABLES USED IN THE PROGRAM:

* A1 -> POPULATION
* A2 -> IMMIGRANTS LAST YEAR
* A3 -> DEATHS LAST YEAR
* B1 -> BUSHELS IN STOREHOUSES
* B2 -> BUSHELS DESTROYED LAST YEAR BY RATS
* B3 -> HARVEST PER ACRE LAST YEAR
* B4 -> TOTAL HARVEST LAST YEAR
* C1 -> ACRES OWNED BY CITY
* C2 -> VALUE OF AN ACRE
* I -> TEMPORARY VARIABLE
* J ->

* INITIALIZATION

250 PRINT 'HAMURABI - '
260 PRINT 'WHERE YOU GOVERN THE ANCIENT KINGDOM OF SUMERIA.'
270 PRINT 'THE OBJECT IS TO FIGURE OUT HOW THE GAME WORKS!!'
280 PRINT 'IF YOU WANT TO QUIT, SELL ALL OF YOUR LAND.'

290 PRINT

INPUT 'ARE YOU READY TO BEGIN? (TYPE Y IF YOU ARE): ' X\$

IF X\$ = 'Y' GOTO 400

400 A1=100

410 A2=5

420 A3=0

430 B1=2000

440 B2=200

450 B3=3

460 B4=3000

470 C1=1000

480 J=1

* REPORT TO HAMURABI

1010 ERASE

1020 PRINT 'HAMURABI, I BEG TO REPORT THAT LAST YEAR'

1030 PRINT

1040 PRINT '#A3' PEOPLE STARVED AND '#A2' PEOPLE CAME TO THE CITY.'

1050 IF J=0 GOTO 1100

1060 A1=A1-INT(A1/2)

1070 PRINT 'THE PLAGUE KILLED HALF THE PEOPLE.'

1100 PRINT 'THE POPULATION IS NOW '#A1'.'

1110 PRINT

1120 PRINT 'WE HARVESTED '#B4' BUSHELS AT '#B3' BUSHELS PER ACRE.'

1130 PRINT 'RATS DESTROYED '#B2' BUSHELS, LEAVING '#B1

1140 PRINT 'BUSHELS IN THE STOREHOUSES.'

1150 PRINT

1160 PRINT 'THE CITY OWNS '#C1' ACRES OF LAND.'

1170 C2=17+INT(ABS(6*RND(-1)))

1180 PRINT 'LAND IS WORTH '#C2' BUSHELS PER ACRE.'

INPUT 'HAVE YOU READ YOUR REPORT, HAMURABI? (TYPE Y IF DONE): ' X\$

IF X\$ = 'Y' GOTO 1185

1185 ERASE

1190 PRINT

1200 PRINT

1210 PRINT 'HAMURABI . . . '

* BUY LAND

1310 PRINT

1320 PRINT 'BUY HOW MANY ACRES?':

1330 INPUT I

1340 I=INT(ABS(I))
1350 IF I=0 GOTO 1510
1360 J=I+C2
1370 IF J<= B1 GOTO 1400
1380 GOSUB 9005
1390 GOTO 1310
1400 B1=B1-J
1410 C1=C1+I
* SELL LAND

1510 PRINT 'SELL HOW MANY ACRES?'
1520 INPUT I
1530 I=INT(ABS(I))
1540 IF I=0 GOTO 1710
1550 IF I>C1 GOTO 1600
1560 IF I<C1 GOTO 9999
1570 GOSUB 9000
1580 GOTO 1510
1600 C1=C1-I
1610 B1=B1+C2*I
* DISTRIBUTE LAND
1710 PRINT 'HOW MANY BUSHELS SHALL WE DISTRIBUTE AS FOOD?':
1720 INPUT I
1730 I=INT(ABS(I))
1740 IF I<=B1 GOTO 1770
1750 GOSUB 9005

1760 GOTO 1710

1770 B1=B1-I

1780 A3=A1-INT(I/20)

1790 A2=0

1800 IF A3 =0 GOTO 1910

1810 A2=A3/2

1820 A3=0

* PLANT LAND

1910 PRINT 'HOW MANY ACRES SHALL WE PLANT?':

1920 INPUT I

1930 I=INT(ABS(I))

1935 IF I>C1 GOTO 1960

1940 J=INT(I/2)

1950 IF J<=B1 GOTO 1980

1960 GOSUB 9005

1970 GOTO 1910

1980 IF I>10*A1 GOTO 1960

ERASE

* HARVEST, RATS, AND CHANGES IN POPULATION

2010 B3=INT(ABS(5*RND(-1)))+1

2020 B4=B3*I

2030 B2=INT(ABS((B1+B4)*.07*RND(-1)))

2040 B1=B1-B2+B4

2050 J=INT(ABS(10*RND(-1)))

2060 A2=INT(A2+(5-B3)*B1/600+1)

2070 IF A2<=50 GOTO 2100

2080 A2=50

2100 A1=A1+A2-A3

2110 GOTO 1010

* ERROR SUBROUTINE

9005 PRINT

9010 PRINT 'HAMURABI, THINK AGAIN - YOU ONLY HAVE'

9020 PRINT '#A1' PEOPLE, '#C1' ACRES, AND':

9030 PRINT '#B1' BUSHELS IN STOREHOUSES.'

9040 PRINT

9050 RETURN

9999 STOP

```

* CHASE GAME BY KIM HERRICK
ERASE
PRINT! THIS IS THE GAME OF CHASE!
10 INPUT! DO YOU WANT INSTRUCTIONS? (TYPE 1 FOR YES)!Y
IF Y<>1 GOTO 20
ERASE
PRINT! YOU ARE "X" IN A HIGH VOLTAGE MAZE WITH 5 MALFUNCTIONING!
PRINT! ROBOTS "O", TRYING TO DESTROY YOU. EASY TIME YOU MAKE!
PRINT! A MOVE, THEY EACH TAKE ONE STEP TOWARD YOU. YOU MUST MANUEVER!
PRINT! EACH OF THEM INTO THE MAZE "X", TO SURVIVE. GOOD LUCK!
PRINT
PRINT! MOVES ARE: 3 2 1!
PRINT! 4 * 8 0 IS NO MOVE!
PRINT! 5 6 7 9 TO QUIT!
PRINT
INPUT! READY? (PUSH RETURN)!Y
20 ERASE
DIM A(200)*E(21)*F(21)
* INITIALIZE ARRAY WITH ZEROS
FOR K=1 TO 200
A(K)=0
PRINT! 1:25:COUNTDOWN: 2:200-N
NEXT K
ERASE
PRINT! 5:55:MOVES:
PRINT! 6:55:3 2 1!
PRINT! 7:55:4 * 8 0 TO STAND STILL!
PRINT! 8:55:5 6 7 9 TO QUIT!
* PRINT FIELD: B IS ROW NUMBER; C IS COLUMN NUMBER
C=1
GOSUB 220
C=20
GOSUB 220
B=1
GOSUB 240
B=10
GOSUB 240
* LOOP TO GENERATE RANDOM LOCATIONS OF PLAYERS
* 5 ROBOTS, 1 PLAYER, 15 HIGH VOLTAGE POLES
25 FOR D=1 TO 21
* COMPENSATION FOR BAD RANDOM NUMBER GENERATOR!!!
30 B=INT(2*(7*(FAC(10000+RND(1))))))
C=INT(2*(17*(FAC(10000+RND(1))))))
N=(10*(C-1))+B
IF A(N)<>0 GOTO 30
IF D>6 A(N)=1
IF D<6 LET A(N)=2
IF D=6 LET A(N)=3
E(D)=B
F(D)=C
IF A(N)=1 PRINT! B+3+C+2+10 'X'
IF A(N)=2 PRINT! B+3+C+2+10 '!'
IF A(N)=3 PRINT! B+3+C+2+10 'O'
NEXT D
* NOW WE MAKE OUR MOVES
G=0
K6=F
50 PRINT! 15:5
INPUT! YOUR MOVE?!Y
*ERASE "YOUR MOVE"
PRINT! 16:5!
B=E(K6)
C=F(K6)
N=10*(C-1)+B
A(N)=0
PRINT! B+3+C+2+10 '!'

```

```

IF Y>8 GOTO 320
IF Y=0 GOTO 100
IF Y<4 GOTO 60
IF Y=8 GOTO 80
IF Y>4 GOTO 70
GOTO 80
60 B=B-1
GOTO 80
70 B=B+1
80 GOTO 80+Y
81 C=C+1
82 GOTO 100
* Z=1 IS A DUMMY STATEMENT TO ALLOW AN ADDRESS; ACTION IS ON NEXT LINE-
83 Z=1
84 Z=1
85 C=C-1
86 GOTO 100
87 Z=1
88 C=C+1
100 N=(10*(C-1))+B
IF A(N)=1 GOTO 320
IF A(N)=2 GOTO 330
A(N)=3
E(N6)=B
F(N6)=C
PRINT! B+3+C+2+10 '!'
* RESET ROBOT LOCATIONS
FOR D=1 TO 5
N=(10*(F(D)-1))+E(D)
IF A(N)<>2 GOTO 200
A(N)=0
PRINT! E(D)+3+F(D)+2+10 '!'
IF E(D)<4 LET E(D)=E(D)+1
IF E(D)>8 LET E(D)=E(D)-1
IF F(D)<4 LET F(D)=F(D)+1
IF F(D)>8 LET F(D)=F(D)-1
N=(10*(F(D)-1))+E(D)
IF A(N)=1 B=B+1
IF A(N)=2 B=B-1
IF A(N)=3 GOTO 330
IF A(N)=0 LET A(N)=2
IF A(N)=2 PRINT! E(D)+3+F(D)+2+10 '!'
IF B=5 GOTO 340
200 NEXT D
GOTO 50
220 FOR B=1 TO 10
N=10*(C-1)+B
A(N)=1
PRINT! B+3+2+C+10 'X'
NEXT B
RETURN
240 FOR C=2 TO 19
N=10*(C-1)+B
A(N)=1
PRINT! B+3+2+C+10 'X'
NEXT C
RETURN
300 PRINT! 1:5 'SORRY TO SEE YOU QUIT!'
GOTO 500
320 PRINT! 1:5 'OH! YOU TOUCHED A HIGH VOLTAGE WIRE AND WERE ZAPPED!'
GOTO 500
330 PRINT! 1:5 'ONE OF THE MAD ROBOTS GOT YOU. NOW YOU'VE HAD IT!'
GOTO 500
340 PRINT! 1:5 'GOOD WORK! YOU GOT ALL THE ROBOTS TO DESTROY THEMSELVES.'
500 INPUT! DO YOU WANT TO TRY AGAIN? TYPE 1 FOR YES, 1Y
IF Y=1 GOTO 600

```

```

PRINT! 600:BY! HOPE YOU HAD FUN!
PRINT! 15:0
STOP
600 PRINT! 0:1
PRINT
PRINT
FOR D=1 TO 21
B=E(D)
C=F(D)
N=10*(C-1)+B
E(D)=0
F(D)=0
A(N)=0
PRINT! B+3+C+2+10 '!'
NEXT D
GOTO 25

```

PHAT

- THIS IS A PROGRAM TO ALLOW THE LOADING OF ANY SOURCE (PROGRAM) TAPE INTO MEMORY REGARDLESS OF ANY DUPE
- THAT WOULD NORMALLY CAUSE LOADING ERRORS. JUST GIVE
- THE PROGRAM THE ADDRESS OF THE START OF TAP BUFFER
- AND START THE TAPE HEADS. YOU MUST PUSH THE SYSTEM
- RESET BUTTON TO STOP THE PROGRAM AND THEN EDIT THE
- ADDRESS IN THE FIRST AND LAST LINES AND ANY OTHER
- ERRORS YOU FIND. WHEN A SEMICOLON (;) IS SEEN, THE
- PROGRAM GOES INTO A ROUTINE TO IGNORE IT AND COPY THE
- SOURCE AND PROGRAM NAME INFO ON THE TAPE. THE TEXT
- IS DISPLAYED ON THE SCREEN WITH NO STOPS FOR
- READABILITY AND THERE IS A LITTLE GARBAGE LEFT ON THE
- SCREEN AFTER A SEMICOLON IS SEEN BUT THE MEMORY IS
- CLEAN.

- FORCE LOAD PROGRAM - TO LOAD BAD TAPES INTO MEMORY
- AND DISPLAY ON SCREEN. NINE MESSAGE 3-23-78

0017	0000	R1	EDU	1	
0018	0000	R2	EDU	2	
0019	0000	R3	EDU	3	
0020	0000	R4	EDU	4	
0021	0000	R5	EDU	5	
0022	0000	R6	EDU	6	
0023	0000	R7	EDU	7	
0024	0000	R8	EDU	8	
0025	0000	R9	EDU	9	
0026	0000	MCPR	EDU	396	
0027	0000	MSBT	EDU	6A	
0028	0000	SEMI	EDU	2E9	
0029	0000	LFCH	EDU	24	
0030	0000	ADDR	EDU	185	
0031	0000	ADD1	EDU	17FA	
0032	0000	ADD2	EDU	17FB	
0033	0000	COND	EDU	3C9	
0034	1000	ORG		16P0	
0035	1000 05 FF	LODI>R1		FF	GET MESSAGE
0036	1002 0D 36 5D MORE	LDA>R1	MSBT		
0037	1005 18 06	SECT>ED	ADRS		UNTILL NULL CHAR
0038	1007 C3	SINZ>R3			
0039	1009 3F 03 96	BSTA>UN	MCMR		AND WRITE IT
0040	100B 10 75	MCIR>UN	MORE		TILL IT'S DONE
0041	100D 3F 01 85 ADRS	BSTA>UN	ADDR		GET ADDRESS FOR DATA STORAGE
0042	1010 00 24	ZMSR	LFCH		
0043	1012 05 0	LODI>R1	B		TURN ON TAPE DRIVE
0044	1014 FT	MS7D>R1			
0045	1016 3F 02 E9 INPUT	BSTA>UN	SERI		LOOK FOR START OF DATA
0046	1018 E7 36	COAI>R3	;;		
0047	101A 18 23	BC7A>E0	BLNSRT		
0048	101C 47 7F	ANDI>R3	7F		CLEAR MSW
0049	101E 03	LOBZ>R3			
0050	1020 C2	STKZ>R2			SAVE DATA
0051	1022 CF 97 FA	STNA>R3	ADD1		LOAD IT INTO MEMORY
0052	1024 3F 03 96	BSTA>UN	MCMR		WRITE DATA ON THE SCREEN
0053	1026 E0 2D	COAI>R2	2D		CM FOR RETURN
0054	1028 3C 00 24	BSTA>ED	LFCH		
0055	102A 00 17 FB	LOGA>R2	ADD2		INCREMENT ADDRESS
0056	102C 00 01	ADDI>R2	1		
0057	102E 0E 17 FB	STNA>R2	00C3		
0058	1030 00 00	MCPR>E0	INPUT		
0059	1032 00 17 FB	LODI>R2	ADD1		
0060	1034 04 01	ADDI>R2	1		
0061	1036 CF 17 FA	STNA>R2	ADD1		
0062	1038 30 5E	BSTA>UN	INPUT		
0063	103A 2D 17 FB	LODI>R1	ADD2		HACUP ONE ADDRESS
0064	1042 45 01	SU:R1	1		ERASE LAST SUNCH PRGR MEMORY

0065	1044 C0 17 FB	STNA>R1	ADD2
0066	1047 77 00	PPSL	MC
0067	1049 0D 17 FA	LODI>R1	ADD1
0068	104C A5 00	SAMI>R1	0
0069	104E C0 17 FA	STNA>R1	ADD1
0070	1051 70 00	CPSL	MC
0071	1053 05 00	LODI>R1	9
0072	1055 3F 02 E9 SKIP	BSTA>UN	SERI
0073	1058 F0 70	ODMR>R1	SHIP
0074	105A 1F 10 15	GCIA>UN	INPUT
0075	105D 4C 4F 41 MSBT	ALIT	GET MORE INPUT
0076	1075 00	DATA	'LOAD TAPE INTO MEMORY AT'
			0

SKIP OVER NAME AND SUNCH

6-19-78

WHEN CENTRAL DATA SENT OUT THE EDITOR/ASSEMBLER CORRECTION SHEET AT THE BEGINNING OF THE YEAR TWO SUGGESTIONS I HAD MADE IN THE "2052 COMPUTER USERS NOTES" NEWSLETTER WERE LEFT OUT. SINCE THEN I HAVE ADDED A MODIFICATION TO ALLOW THE ASSEMBLER PROGRAM TO CONTROL THE TAPE RECORDER (IF THE REMOTE CONTROL IS WIRED INTO YOUR SYSTEM). THESE ARE THOSE CHANGES MADE TO WORK WITH ALL PUBLISHED CENTRAL DATA MODIFICATIONS: (INCIDENTALLY, JEFF KOLOFF TELLS ME THE NEW BASIC AND NEW ASSEMBLER PROGRAMS WILL CONTROL THE TAPE RECORDERS SO IT'S WORTH IT TO BUILD THIS INTO YOUR SYSTEM.)

33T5:	04 00 F0 17 30 7A 3F 21 FE 17 30 74 07 5D 1F 26
	30 30 80 3F 2E 32 17 04 24 05 FE 06 92 F0 7E F9
	7E FA 7A 17 04 00 F0 17 30 7A 3F 26 32 17 30 70
	30 65 17 30 0F 3F 01 70 17 30 70 3F 23 16 17 30
	72 30 04 3F 32 10 17
200D:	3F 33 70
2653:	1F 33 7F
26C8:	3F 33 26
29MC:	3F 33 30
250C:	3F 33 43
200F:	3F 33 0E
317D:	3F 33 04

NINE MESSAGE


```

0001      PRNT
0002      TAPE
0003      *
0004      * 10.23 78 CGM FOR X*
0005      *
0006      * SMART TERMINAL PROGRAM
0007      * NO LCRO OF KEYBOARD INPUT
0008      * LINKS TWO COMPUTERS
0009      * TOGETHER VIA FLAG AND SENSE
0010      *
0011      * REGISTER EQUATES
0012      R2 EQU 2
0013      R1 EQU 1
0014      R2 EQU 2
0015      R3 EQU 3
0016      *
0017      EQ EQU 0
0018      LT EQU 2
0019      UN EQU 3
0020      *
0021      * SUPERVISOR ROUTINES CALLED
0022      LFCR EQU 0004 LINEFEED CARRIAGE ROUTINE
0023      RETU EQU 0005 SUPERVISOR RETURN POINT
0024      WCHR EQU 0006 WRITE CHARACTER ROUTINE 000
0025      *
0026      *
0027      SMART LODI,R0 40 SET FLAG ONLY
0028      LPSU EQU 27 DO IT TO UPPER
0029      LODI,R0 27 SET LOGICAL ONLY
0030      LPSU EQU 27 TO IT TO LOWER
0031      LODI,R0 DE 300 BAUD -66 600 BAUD
0032      STRA,R0 BAUD USE LOCAL BAUD RATE
0033      *
0034      READ LODI,R3 00 CLEAR CHAR BUFFFP
0035      LODI,R2 04 SET FOR 8 BITS PER CHAR
0036      TFSU EQU 06 TEST SENSE BIT
0037      BTR,EC SENSIN IF IT DROPPED, BRANCH
0038      RDD,R3 RSLF TEST FOR KEYBOARD INPUT
0039      TMI,R3 06 TEST THE STROBE BIT
0040      BCFR,EC KEYIN IF IT DROPPED, BRANCH
0041      ECTR,UN READ RSLF. LOOP UNTIL INPUT FOUND
0042      *
0043      SENSIN BETA,UN DELY1 GO WAIT 1/2 BIT TIME
0044      TFSU EQU 06 SET IF STILL DROPPED START
0045      RETC,EC RTAD IF NOT RETURN NOISE!
0046      *
0047      NEXT BSTR,UN DELY2 WAIT UNTIL MIDDLE OF BIT
0048      SFSU EQU 06 GET SENSE INPUT R0
0049      ANDI,R0 80 MASK OFF TRASH
0050      RRR,R3 ROTATE CURRENT BIT PATTERN
0051      ICRZ,R3 ADD IN CUR NEW BIT
0052      STEZ,R3 RESTORE IT TO ITS BUFFFP
0053      BDRR,R2 NEXT SEE IF WE GOT IT ALL ELSE LOOP
0054      BSTR,UN DELY1 GO WAIT 1/2 BIT TIME
0055      CCPI,R3 0A TRAP LINE FEED HERE

```

FILE PROGRAM AS PRINTED BY SYSTEM ON 01-19-79

PAGE 0202

```

0056      RETC,EC JUST RETURN IF SO
0057      COMI,R3 0D TRAP CARRIAGE RETURN HERE
0058      BCTA,EC LFCR DO ONE IF FOUND
0059      COMI,R3 1E WAS IT 'ESCAPE'?
0060      BCTR,EC CLEAR TRAP CLEAR SCREEN- CODE
0061      BCTA,UN WCHR IF SO, DO IT
0062      *
0063      KEYIN BRED,R1 KEYBOARD INPUT
0064      TMI,R1 02 TEST STROBE BIT
0065      BCFR,EC KEYIN WAIT FOR IT TO DROP
0066      COMI,R3 1E WAS IT 'ESCAPE'?
0067      BCTA,EC RETU YES, RETURN TO SUPERVISOR
0068      COMI,R3 1E TRY OUR -CLEAR SCREEN- CODE
0069      BCTR,EC CLEAR IF SO, DO IT
0070      *
0071      LODI,R2 8 SET HF INDEX FOR 8 BITS
0072      CPSU EQU 40 DROP THE FLAG FOR START
0073      *
0074      NITBIT BSTR,UN DELY2 GO WAIT ONE BIT TIME
0075      RRR,R3 ROTATE OUR CHARACTER PATTERN
0076      BCTR,LT ONE WAS HIGH BIT A ONE?
0077      CPSU EQU 40 NO, THEN SET THE FLAG FOR 0
0078      BCTR,UN ZERO AND DO A ZERO
0079      SFSU EQU 40 YOU GET THERE FOR A ONE
0080      ZERO BDRR,R2 NITBIT HAVI WE DONE 2 BITS?, NO, LOOP
0081      BSTR,UN DELY2 ELSE, GO WAIT FOR PARITY BIT
0082      SFSU EQU 40 SET FLAG FOR STOP BITS
0083      BSTR,UN DELY4 GO WAIT 2 BIT TIMES
0084      RETC,UN ALL DONE, GET NEXT INPUT
0085      *
0086      DELY2 LODR,R2 BAUD GET LOCAL BAUD RATE
0087      BTR,EC $ WAIT HERE AND COUNT
0088      DELY1 LODR,R2 BAUD GET LOCAL BAUD RATE
0089      BDRR,R2 $ WAIT HERE AND COUNT
0090      RETC,UN ALL DONE
0091      DELY4 BSTR,UN DELY2 DO THE ONE BIT WAIT
0092      BSTR,UN DELY2 DO IT TWICE
0093      RETC,UN ALL DONE
0094      *
0095      *
0096      BAUD RES 1 LOCAL BAUD RATE STORAGE
0097      *
0098      CLEAR ICRZ,R2 CLEAR SCREEN ROUTINE
0099      STRR,R0 *LINE CLEAR LOWER CURSOR POINTER
0100      BSTR,UN CLEND CLEAR LINE @ TC 16
0101      ECTR,UN LFCR LINE FEED BACK TO TOP
0102      RETC,UN ALL DONE
0103      *
0104      *
0105      CLEND LODR,R2 *LINE GET LINE DATA BYTE
0106      TMI,R2 2F SEE IF LINE 16
0107      RETC,EC IF SO, WE ARE DONE
0108      BTR,UN LFCR ELSE, DO A LINE FEED
0109      BCTR,UN CLEND AND GO CHECK LINE COUNT
0110      *
0111      *
0112      LINE ACON 17FF LOWER CURSOR BYTE POINTER
0113      END

```

PRINT

```

0002 0000 *HELLO THIS IS A TEST TRANSMISSION OF THE MORSE CODE
0003 0000 *I HOPE THAT YOU CAN READ IT
0004 0000 *THE QUICK BROWN FOX
0005 0000 *JUMPS OVER THE
0006 0000 *LAZY DOG
0007 0000 *...//---,,,
0008 0000
0009 0000
0010 0000

```

Morse Code Program by Mike Durham

```

0011 0000 *
0012 0000 * HORSE CODE *
0013 0000 * PROGRAM BY MIKE DURHAM *
0014 0000 *
0015 0000 R0 EQU 0
0016 0000 R1 EQU 1
0017 0000 R2 EQU 2
0018 0000 R3 EQU 3
0019 0000 E0 EQU 0
0020 0000 GT EQU 1
0021 0000 LT EQU 2
0022 0000 UN EQU 3
0023 0000 UC EQU 8
0024 0000 L1 EQU 1
0025 0000 L2 EQU 2
0026 0000 L3 EQU 3
0027 0000 L4 EQU 4
0028 0000 L EQU 4C
0029 0000 D EQU 44
0030 0000 CR EQU 0D
0031 0000 *
0032 0000 * SUBROUTINES USED IN SUPERVISOR *
0033 0000 *
0034 0000 WRBL EQU #170
0035 0000 SERI EQU 2E9 DATA -> R3
0036 0000 WCHR EQU 396 WRITES CHARACTER THAT'S IN R3
0037 0000 SERO EQU 24F DATA MUST BE IN R3
0038 0000 INHX EQU #1B6 DATA -> R3
0039 0000 HXOT EQU #06A DATA MUST BE IN R2
0040 0000 LFCR EQU #024
0041 0000 KBIN EQU 30F GETS DATA FROM KEYBOARD -> R3
0042 0000 CHND EQU #3C0
0043 0000
0044 0000 BLKS
0045 0000 TAPE
0046 3500 ORG 3500
0047 3500 1B 02 START BCTR,UN STRT SKIP PINTERS
0048 3502 *
0049 3502 00 TIME1 RES 1
0050 3503 00 TIME3 RES 1
0051 3504 *
0052 3504 3F 35 F3 STRT BSTA,UN ERASE
0053 3507 75 10 CPSL 10 SELECT REGISTER BANK #0
0054 3509 05 00 LODI,R1 #0
0055 350B C0 15 BA STRA,R1 TEMP
0056 350E C0 15 BC STRA,R1 TEMP3
0057 3511 C0 15 B7 STRA,R1 PTR2 ZERO IT
0058 3514 0D 95 88 LODA,R1 #TXTPTR
0059 3517 C0 15 06 STRA,R1 PTR1 RESET STRT OF TEXT POINTER
0060 351A 05 07 LODI,R1 #7
0061 351C F1 WRTO,R1 SET FLAG TO 0
0062 351D 3F 35 FB BSTA,UN GETINE GET MORSE RATE FROM USER
0063 3520 3F 35 F3 BSTA,UN ERASE
0064 3523 71 RESTRT REDD,R1 READ K/BOARD PORT
0065 3524 F5 00 TH1,R1 #0 TEST STROBE FROM K/BOARD
0066 3526 9C 36 1A BCFA,E0 DELAY
0067 3529 3F 35 0D BSTA,UN GETC GET CHARACTER TO BE SENT INTO R3
0068 352C 3F 35 BA BSTA,UN LOOKUP GET MORSE CODE & LENGTH INTO R2 & R0
0069 352F 44 F8 ANDI,R0 MASK OFF LENGTH
0070 3531 46 07 ANDI,R2 #7 MASK OFF CHAR LEAVING LENGTH
0071 3533 3F 35 3B BSTA,UN SENDIT SEND OUT CHAR
0072 3536 3F 35 C1 BSTA,UN GAP PUT A GAP BETWEEN CHARACTERS
0073 3539 1B 68 BCTR,UN RESTRT DO IT AGAIN SAM
0074 353B *
0075 353B * SEND OUT CHAR IN R0 / LENGTH IS IN R2.
0076 353B *
0077 353B 68 SENDIT IORZ,R0 MAKE CC = CONTENTS OF R0
0078 353C 1E 35 49 BCTA,LT DIT SEND DIT IF MSB = 1
0079 353F 1F 35 5C BCTA,UN DAH SEND DAH IF MSB = 0
0080 3542 3F 35 B0 RET BSTA,UN PAUSE DO A PAUSE BETWEEN BITS & DAHS
0081 3545 D0 RRL,R0 MOVE NEXT BIT TO MSB OF R0
0082 3546 FA 73 BDRR,R2 SENDIT DO AGAIN IF MORE BITS TO COME
0083 3548 17 RETC,UN RESTRT GET NEXT CHARACTER
0084 3549 *
0085 3549 07 0F DIT LODI,R3 #F
0086 354B F3 WRTO,R3 SET FLAG TO 1
0087 354C 0F 15 02 LODA,R3 TIME1 GET DELAY CONSTANT
0088 354F 0D 15 02 LODA,R1 TIME1
0089 3552 FB 7E LOOP1 BDRR,R3 LOOP1
0090 3554 F9 7C BDRR,R1 LOOP1
0091 3556 07 07 LODI,R3 #7
0092 3558 F3 WRTO,R3 FLAG TO OFF
0093 3559 1F 35 42 BCTA,UN RET
0094 355C *
0095 355C 07 0F DAH LODI,R3 #F SET FLAG TO 1
0096 355E F3 WRTO,R3
0097 355F 77 10 PPSL 10
0098 3561 06 03 LODI,R2 #3
0099 3563 0F 15 02 LITLOP LODA,R3 TIME1
0100 3566 0D 15 02 LODA,R1 TIME1
0101 3569 FB 7E LDOP2 BDRR,R3 LOOP2

```

0102	3568	F9 7C	BDRR,R1	LOOP2	
0103	356D	FA 74	BDRR,R2	LITLOP	
0104	356E	07 07	LODI,R3	07	
0105	3571	F3	WRTO,R3		FLAG TO 0
0106	3572	75 10	CPSL	10	
0107	3574	1F 35 42	BCTA,UN	RET	
0108	3577	*			
0109	3577	09 0E	INC	LODR,R1	PTR2
0110	3579	85 01		ADDI,R1	01
0111	357B	C9 0A		STRR,R1	PTR2
0112	357D	15		RETC,GT	
0113	357E	16		RETC,LT	
0114	357F	09 05		LODR,R1	PTR1
0115	3581	85 01		ADDI,R1	01
0116	3583	C9 01		STRR,R1	PTR1
0117	3585	17		RETC,UN	
0118	3586	*			
0119	3586	*			
0120	3586	00	PTR1	RES	1 POINTS TO TEXT
0121	3587	00	PTR2	RES	1
0122	3588	*			
0123	3588	20 03	TXTPTR	ACON	2003
0124	358A	*			
0125	358A	*	GETC	GETS VALID CHAR FROM TEXT INTO R3 (0->9 OR A->Z)	
0126	358A	*			
0127	358A	*			
0128	358A	00	TEMP	RES	1
0129	358B	00	TEMP2	RES	1
0130	358C	00	TEMP3	RES	1
0131	358D	0B F7	GETC	LODR,R3	*PTR1 GET CHARACTER FROM TEXT
0132	358F	3F 35 77		BSTA,UN	INC
0133	3592	E7 20		COMI,R3	20 SPACE
0134	3594	1C 35 DB		BCTA,ED	SPACE
0135	3597	E7 00		COMI,R3	00 ENBLINE
0136	3599	1C 35 E7		BCTA,ED	ENDLIN
0137	359C	E7 2C		COMI,R3	2C
0138	359E	1E 35 23		BCTA,LT	RESTR
0139	35A1	E7 5A		COMI,R3	5A ASCII 7
0140	35A3	1D 35 23		BCTA,GT	RESTR
0141	35A6	E7 41		COMI,R3	41 ASCII A
0142	35AB	3A 0A		BSTR,LT	CHECK MAKE SURE ITS A NUMBER
0143	35AA	CF 15 0B		STRA,R3	TEMP2 SAVE CHAR
0144	35AD	3F 03 96		BSTA,UN	VCHR
0145	35B0	0F 15 0B		LODA,R3	TEMP2 RESTORE R3
0146	35B3	17		RETC,UN	RETURN WITH VALID CHARACTER
0147	35B4	*			
0148	35B4	E7 39	CHECK	COMI,R3	39 ASCII 9
0149	35B6	1D 35 23		BCTA,0T	RESTR
0150	35B9	17		RETC,UN	GO BACK IF CHAR IS ILLEGAL RETURN WITH VALID CHARACTER
0151	35BA	*			
0152	35BA	*			
0153	35BA	*			
0154	35BA	*			
0155	35BA	A7 2C	LOOKUP	SUBI,R3	2C PREPARE INDEX REG
0156	35BC	0F 76 92		LODA,R3	TABLE,I GET CODE FROM TABLE R3 = INDEX REG
0157	35BF	C2		STR2,R2	
0158	35C0	17		RETC,UN	
0159	35C1	*			
0160	35C1	*			
0161	35C1	*			
0162	35C1	*			
0163	35C1	06 04	GAP	LOBI,R2	04
0164	35C3	0F 15 03	XX	LOBA,R3	TIME3
0165	35C6	0D 15 03		LODA,R1	TIME3
0166	35C9	F8 7E	LOOP3	BDRR,R3	LOOP3
0167	35CB	F9 7C		BDRR,R1	LBOP3
0168	35CD	FA 74		BDRR,R2	XX
0169	35CF	17		RETC,UN	
0170	35D0	*			
0171	35D0	*			
0172	35D0	*			
0173	35D0	*			
0174	35D0	0F 15 02	PAUSE	LODA,R3	TIME1
0175	35D3	0B 15 02		LODA,R1	TIME1
0176	35D6	F8 7E	LOOP4	BDRR,R3	LOOP4
0177	35D8	F9 7C		BDRR,R1	LOOP4
0178	35DA	17		RETC,UN	
0179	35DB	*			
0180	35DB	*			
0181	35DB	*			
0182	35DB	3F 35 C1	SPACE	BSTA,UN	GAP
0183	35DE	3F 35 C1		BSTA,UN	GAP
0184	35E1	3F 01 70		BSTA,UN	WRBL
0185	35E4	1F 35 23		BCTA,UN	RESTR
0186	35E7	*			
0187	35E7	3F 35 C1	ENDLIN	BSTA,UN	GAP
0188	35EA	3F 35 C1		BSTA,UN	GAP
0189	35EB	3F 00 24		BSTA,UN	LFCR
0190	35F0	1F 35 23		BCTA,UN	RESTR
0191	35F3	*			
0192	35F3	*			
0193	35F3	05 10	ERASE	LODI,R1	10
0194	35F5	3F 00 24	LOOP5	BSTA,UN	LFCR
0195	35F8	F9 7B		BDRR,R1	LOOP5
0196	35FA	17		RETC,UN	
0197	35FB	*			
0198	35FB	3F 36 27	GETIME	BSTA,UN	PRINT LENGTH OF BIT
0199	35FE	3F 01 06		BSTA,UN	INMX GET LENGTH
0200	3601	CF 15 02		STRA,R3	TIME1
0201	3604	3F 00 24		BSTA,UN	LFCR
0202	3607	*			
0203	3607	*			

0204	3607	3F	36	27	BSTA,UN	PRINT	INPUT PAUSE BETWEEN LETTERS
0205	360A	3F	01	06	BSTA,UN	INHX	GET IT
0206	360D	CF	15	03	STRA,R3	TIMES	
0207	3610	3F	00	24	BSTA,UN	LFCR	
0208	3613						
0209	3613	3F	36	27	BSTA,UN	PRINT	PRESS ANY KEY TO GO
0210	3616	3F	03	0F	BSTA,UN	KBIN	
0211	3619	17			RETC,UN		
0212	361A						
0213	361A						* DO A SMALL DELAY
0214	361A						
0215	361A	20			DELAY	EORZ,R0	
0216	361B	C1				STRZ,R1	
0217	361C	06	02			LODI,R2	02
0218	361E	F8	7E		ZZ	BDRR,R0	ZZ
0219	3620	F9	7C			BDRR,R1	ZZ
0220	3622	FA	7A			BDRR,R2	ZZ
0221	3624	1F	35	00		BCTA,UN	START
0222	3627						
0223	3627						
0224	3627	0D	15	0C	PRINT	LODA,R1	TEMP3 GET INDEX
0225	362A	0D	36	39		LODA,R1	INFO,+ GET CHAR TO BE PRINTED
0226	362D	CD	15	0C		STRA,R1	TEMP3
0227	3630	E4	2C			COMI,R0	2C , ,
0228	3632	14				RETC,EQ	
0229	3633	C3				STRZ,R3	
0230	3634	3F	03	96	BSTA,UN	MCHR	WRITE IT ON SCREEN
0231	3637	1B	6E		BCTR,UN	PRINT	DO IT AGAIN
0232	3639						
0233	3639	20	20	20	INFO	ALIT	INPUT LENGTH OF DIT :- ,
0234	365A	20	49	4E		ALIT	INPUT SPACE BETWEEN LETTERS :- , PRESS ANY KEY TO GO.,
0235	3692						
0236	3692	36			TABLE	DATA	36 ,
0237	3693	7E				DATA	7E -
0238	3694	AE				DATA	AE .
0239	3695	6D				DATA	6D /
0240	3696	05				DATA	05 0
0241	3697	85				DATA	85 1
0242	3698	C5				DATA	C5 2
0243	3699	E5				DATA	E5 3
0244	369A	F5				DATA	F5 4
0245	369B	FD				DATA	FD 5
0246	369C	7D				DATA	7D 6
0247	369D	3D				DATA	3D 7
0248	369E	1D				DATA	1D 8
0249	369F	0D				DATA	0D 9
0250	36A0	00	00	00		RES	7
0251	36A7				****		
0252	36A7	82				DATA	82 A
0253	36A8	74				DATA	74 B
0254	36A9	54				DATA	54 C
0255	36AA	63				DATA	63 D
0256	36AB	B1				DATA	B1 E
0257	36AC	D4				DATA	D4 F
0258	36AD	23				DATA	23 G
0259	36AE	F4				DATA	F4 H
0260	36AF	C2				DATA	C2 I
0261	36B0	B4				DATA	B4 J
0262	36B1	43				DATA	43 K
0263	36B2	B4				DATA	B4 L
0264	36B3	02				DATA	02 M
0265	36B4	42				DATA	42 N
0266	36B5	03				DATA	03 O
0267	36B6	94				DATA	94 P
0268	36B7	24				DATA	24 Q
0269	36B8	A3				DATA	A3 R
0270	36B9	E3				DATA	E3 S
0271	36BA	01				DATA	01 T
0272	36BB	C3				DATA	C3 U
0273	36BC	E4				DATA	E4 V
0274	36BD	83				DATA	83 W
0275	36BE	64				DATA	64 X
0276	36BF	D4				DATA	D4 Y
0277	36C0	34				DATA	34 Z
0278	36C1						

Selectric Driver Routine by Dennis Haynes

FILE 'DUMP' AS ASSEMBLED BY SYSTEM ON 03-04-69

LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND	COMMENTS
0001	0000				*		TEXT	OUTPUT ROUTINE	
0002	0000				*				
0003	0000				*		OUTPUTS	ASCII CHARACTERS FROM MEMORY STARTING	
0004	0 00				*		AT	LOCATION POINTED TO BY PTR	
0005	0000				*		USES	TREN ROUTINE TO PRINT CHR TO PRINTER	
0006	0000				*		LOOKS	FOR ↑ AS AN END OF TEXT CHR	
0007	0000				*		USES	R0 VALUE IS DESTROYED BY TREN ROUTINE	
0008	0000				*		SUBROUTINE	NESTING 1 LEVELS	
0009	0000				XPRNT	EQU	4002	ENTRY POINT	PRINT ROUTINE
0010	0000				DOS	EQU	6004	DOS	RE-ENTRY POINT
0011	0000					EXEC	4200		
0012	4200					ORG	4200		
0013	4200	77	0A			PPSL	WC+COM		
0014	4202	75	01			CPSL	CRY		
0015	4204	0C	82	21	STRT	LODA,R0	*PTR	GET CHR FROM BUFFER	
0016	4207	E4	5E			COMI,R0	'↑'	UP ARROW	
0017	4209	13	11			BCTR,EQ	RET		
0018	420B	3F	40	02		BSTA,UN	XPRNT		
0019	420E	03	12			LODR,R0	PTR+1		
0020	4210	34	01			ADDI,R0	01		
0021	4212	C8	0E			STRR,R0	PTR+1		
0022	4214	08	0B			LODR,R0	PTR		
0023	4216	34	00			ADDI,R0	00		
0024	4218	C8	07			STRR,R0	PTR		
0025	421A	1B	68			BCTR,UN	STRT		
0026	421C	75	03		RET	CPSL	WC		
0027	421E	1F	E0	04		BCTA,UN	*DOS		
0028	4221	44			PTR	DATA	44		
0029	4222	00				DATA	00		

ASCII TO SELECTRIC CONVERSION ROUTINE

This routine converts ASCII character in R0 to the equivalent Selectric Typewriter code and outputs it via a programmable serial IO port to a Trendata terminal.

The IBM Correspondence code is used here but the routine can be used for EBCDIC code as well by changing the look up table data. Both IBM codes use 6 data bits. Since there are only 64 possible combinations of 6 bits there obviously aren't enough unique codes to represent all of the characters one would like to print. This problem is overcome by the use of shift characters. When the selectric printer receives an UpCase character all of the characters which follow are assumed to be upper case until a LwCase character is received.

The computer is responsible for keeping track of the upper/lower case status and sending shift characters as required. To accomplish this a code for upper case or lower case is added to the 6 bit IBM code using the 2 highest order unused bits (bits 6&7). If a character is lower case these 2 bits are 00. If a character is upper case they are 01. A 10 indicates an either case (i.e. "CarRet", ".", etc.).

When the computer looks up the IBM code in the table it checks this code against the code for the last character sent and outputs a shift character where required.

The program contains provisions for a delay following carriage returns and tabs. It also has a provision for implementing a form feed by keeping track of the current line number and adding extra line feeds to advance the paper by the number of lines required to reach the top of the page when a form feed character is received (a very useful function for long assembly listings).

The subroutine nesting is 1 level if the form feed is used otherwise it is 0 levels. The routine could be shortened somewhat by using more subroutines at the expense of decreasing the number of levels of subroutines available to the calling program.

```

LINE ADDR B1 B2 B3 B4 LABEL OPCODE OPERAND COMMENTS
0001 0000 *
0002 0000 *
0003 0000 * DRIVER PROGRAM TO OUTPUT CHARACTERS TO TRENDATA
0004 0000 * TERMINAL IN CORRESPONDENT CODE
0005 0000 *
0006 0000 *
0007 0000 *
0008 0000 *
0009 0000 *
0010 0000 * * PCI BOARD AS OF 11-26-78 HAS INVERTING BUS *
0011 0000 * * XCEVERS HENCE ALL DATA WRITTEN TO PCI *
0012 0000 * * MUST BE COMPLIMENTED PRIOR TO SENDING *
0013 0000 * * ALL B0R1,R1 FF INSTRUCTIONS ARE FOR *
0014 0000 * * THIS PURPOSE AND MUST BE DELETED IF BUFFERS *
0015 0000 * * ARE CHANGED AT A FUTURE DATE *
0016 0000 * *****
0017 0000 *
0018 0000 *
0019 0000 * PCI INITIALIZATION ROUTINE
0020 0000 *
0021 0000 *
0022 0000 * REGISTER EQUATES
0023 0000 *
0024 0000 *
0025 0000 R0 EQU 0
0026 0000 R1 EQU 1
0027 0000 R2 EQU 2
0028 0000 R3 EQU 3
0029 0000 RQ EQU 0
0030 0000 CT EQU 1
0031 0000 LT EQU 2
0032 0000 UN EQU 3
0033 0000 NE EQU 80
0034 0000 LE EQU 91
0035 0000 GE EQU 82
0036 0000 Z EQU 0
0037 0000 N EQU 2
0038 0000 CRY EQU 1 PSL
0039 0000 COM EQU 2
0040 0000 OVP EQU 4
0041 0000 WC EQU 8
0042 0000 RS EQU 10
0043 0000 IDC EQU 20
0044 0000 COND EQU 0
0045 0000 SP EQU 7 PSU
0046 0000 TI EQU 20
0047 0000 FLAG EQU 40
0048 0000 SENSE EQU 80
0049 0000 *
0050 0000 ESC EQU 1B
0051 0000 BS EQU 8
0052 0000 MR1 EQU 02
0053 0000 CR EQU 03
0054 0000 EXEC XINIT AUTOEXEC & INIT ROUTINE
0055 1520 ORG 1520
    
```

```

LINE ADDR B1 B2 B3 B4 LABEL OPCODE OPERAND COMMENTS
0056 1520 1B 02 XINIT BCTR,UN INIT POINTER TO INIT ROUTINE
0057 1522 1B 2F XPRNT BCTR,UN PRINT POINTER TO PRINT ROUTINE
0058 1524 *
0059 1524 * ROUTINE TO INITIALIZE 2651 PROGRAMMABLE COMMUNICATION INTERFACE
0060 1524 *
0061 1524 04 75 INIT LODI,R0 75 1 STOP BIT EVEN PAR,6BIT CHR,1X CLOCK RATE
0062 1526 24 FF EORI,R0 FF
0063 1528 04 02 WRITE,R0 MR1
0064 152A 04 33 LODI,R0 33 INTERNAL CLOCK,134.5 BAUD
0065 152C 24 FF EORI,R0 FF
0066 152E 04 02 WRITE,R0 MR1
0067 1530 04 05 LODI,R0 05 NORMAL OPERATION,RCV & XMIT ENABLED
0068 1532 24 FF EORI,R0 FF
0069 1534 04 03 WRITE,R0 CR
0070 1536 04 00 LODI,R0 00
0071 1538 CC 95 51 STRA,R0 *LNMR ZERO LINE NUMBER COUNT FOR FORM FEED
0072 153B * FOLLOWING ROUTINE IS TO INIT PRINTER TO KNOWN STATE WHEN OPERATING
0073 153B * IN THE CR TEST MODE. MAY NOT BE REQUIRED WITH OTHER SELECTRIC PRINTERS
0074 153B 04 34 LODI,R0 34 SOT CODE
0075 153D 3F 15 D6 BSTA,UN OUT
0076 1540 04 34 LODI,R0 34 SOT
0077 1542 3F 15 D6 BSTA,UN OUT
0078 1545 04 9C LODI,R0 9C LOAD UC SHIFT
0079 1547 3F 15 D6 BSTA,UN OUT
0080 154A 1F E0 04 BCTA,UN *6004 BRANCH TO DOS
0081 154D *
0082 154D * CHARACTER OUTPUT SUBROUTINE
0083 154D *
0084 154D * ASSUMES PCI HAS ALREADY BEEN INITIALIZED
0085 154D * ACCEPTS CHR TO BE PRINTED IN R0
0086 154D * USES REGISTERS R0,R1
0087 154D * RESTORES ORIG. VALUE TO R1, R0 IS DESTROYED
0088 154D * SUBROUTINE NESTING & LEVELS
0089 154D SR EQU 01
0090 154D THR EQU 00
0091 154D TXRDY EQU 01
0092 154D 15 10 RITEM ACON 1510 TEMP STORAGE FOR R1 MAY BE ANY RAM LOC
0093 154F 15 12 SFLAG ACON 1512 LAST CHR SHIFT CODE MAY BE ANY RAM LOC
0094 1551 15 11 LNMR ACON 1511
0095 1553 *
0096 1553 *
0097 1553 E4 0A PRINT COMI,R0 0A IGNORE LINE FEEDS
0098 1555 14 RETC,EQ
0099 1556 E4 0C COMI,R0 0C FORM FEED
0100 1558 98 24 BCFR,EQ F1
0101 155A CC 95 4D STRA,R1 *RITEM SAVE R1
0102 155B CC 95 51 F2 LODI,R0 *LNMR GET LINE NUMBER
0103 1560 04 42 COMI,R0 42 CHECK FOR TOP OF NEXT PAGE
0104 1562 1A 05 BCTR,LT F3 BR IF NOT
0105 1564 20 EORZ,R0
0106 1565 CC 95 51 STRA,R0 *LNMR ZERO LNMR COUNT
0107 1568 14 RETC,EQ RET IF EQ
0108 1569 04 AE F3 LODI,R0 AE CORSP LINE FEED
0109 156B 3F 15 D6 BSTA,UN OUT
0110 156E 04 FF LODI,R0 FF
    
```

FILE 'TRN' AS ASSEMBLED BY SYSTEM ON 03-05-79

LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND	COMMENTS
0111	1570	F9	7E				BDRR, R1	\$	DELAY LOOP FOR LINE FEED
0112	1572	F8	7C				BDRR, R0	\$-2	
0113	1574	04	01				LODI, R0	01	
0114	1576	8C	95	51			ADDA, R0	*LNMR	
0115	1579	CC	95	51			STRA, R0	*LNMR	
0116	157C	1B	5F				BCTR, UN	F2	
0117	157E								THIS PORTION OF ROUTINE CHECKS FOR GENERAL DATA TURN PRINTER ON/OFF
0118	157E								COMMAND AS WELL AS ASCII SOT AND EOT
0119	157E	F4	80			F1	TMI, R0	80	CHECK FOR SOT
0120	1580	18	07				BCTR, EQ	SOT	BR TO ST OF TEXT ROUTINE
0121	1582	E4	03				COMI, R0	03	CHECK FOR SOT
0122	1584	14					RETC, EQ		RET IF EQ
0123	1585	E4	02				COMI, R0	02	
0124	1587	98	0D				BCFR, EQ	*RGRP	CHECK IF ASCII SOT
0125	1589	00				SOT	LODI, R0	00	CORSP SP CODE
0126	158B	CC	95	51			STRA, R0	*LNMR	ZERO LN NUMBER COUNT
0127	158E	3F	15	D6			HSTA, UN	OUT	
0128	1591	04	0C				LODI, R0	9C	UC SHIFT
0129	1593	1F	15	D6			BCTA, UN	OUT	
0130	1596	CD	95	4D		LRUP	STRA, R1	*RITEM	
0131	1599	01					STRA, R1		LOAD ASCII VALUE INTO INDEX REG
0132	159A	0D	76	06			LODA, R1	TABLE, I	GET CORSP CODE
0133	159D	E4	00				COMI, R0	CO	TEST FOR NON-PRINTABLE CHR
0134	159F	98	04				BCFR, EQ	SHIFT	BRANCH IF PRINTABLE
0135	15A1	0D	95	4D			LODA, R1	*RITEM	
0136	15A4	17					RETC, UN		
0137	15A5								THIS ROUTINE CHECKS TO SEE IF A SHIFT IS REQUIRED
0138	15A5	F4	80			SHIFT	TMI, R0	80	
0139	15A7	13	2D				BCTR, EQ	OUT	
0140	15A9	C1					STRA, R1		
0141	15AA	45	40				ANDI, R1	40	
0142	15AC	ED	95	4F			COMA, R1	*SFLAG	COMPARE U.C./L.C CODE TO LAST CHR CODE
0143	15AF	18	25				BCTR, EQ	OUT	OUTPUT CHR IF SHIFT NOT REQ
0144	15B1	CD	95	4F			STRA, R1	*SFLAG	
0145	15B4	F5	40				TMI, R1	40	TEST FOR U.C.
0146	15B5	98	10				BCFR, R0	LC	
0147	15B8	55	01			T2	REDE, R1	SR	
0148	15BA	25	FF				EORI, R1	FF	
0149	15BC	F5	01				TMI, R1	TXRDY	
0150	15BE	98	78				BCFR, EQ	T2	
0151	15C0	05	9C				LODI, R1	9C	SEND U.C. SHIFT CHR
0152	15C2	25	FF				EORI, R1	FF	
0153	15C4	D5	00				WRITE, R1	THR	
0154	15C6	1B	0E				BCTR, UN	OUT	
0155	15C8	55	01			LC	REDE, R1	SR	
0156	15CA	25	FF				EORI, R1	FF	
0157	15CC	F5	01				TMI, R1	TXRDY	
0158	15CE	98	78				BCFR, EQ	LC	
0159	15D0	25	FF				LODI, R1	9F	SEND L.C. SHIFT CHR
0160	15D2	25	FF				EORI, R1	FF	
0161	15D4	D5	00				WRITE, R1	THR	
0162	15D6	55	01			OUT	REDE, R1	SR	WAIT LOOP TIL THR IS EMPTY
0163	15D8	25	FF				EORI, R1	FF	
0164	15DA	F5	01				TMI, R1	TXRDY	
0165	15DC	93	78				BCFR, EQ	OUT	

FILE 'TRN' AS ASSEMBLED BY SYSTEM ON 03-05-79

LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND	COMMENTS
0166	15DE	24	FF				EORI, R0	FF	
0167	15E0	D4	90				WRITE, R0	THR	
0168	15E2	24	FF				EORI, R0	FF	
0169	15E4	B4	AF				COMI, R0	AF	CHECK IF CHR WAS TAB
0170	15E6	18	12				BCTR, EQ	DLY	
0171	15E8	E4	AD				COMI, R0	AD	CHECK FOR CARRAGE RETURN
0172	15EA	98	0A				BCFR, EQ	RST	
0173	15EC	04	01				LODI, R0	01	
0174	15EE	8C	95	51			ADDA, R0	*LNMR	INC LINE NUMBER COUNT
0175	15F1	CC	95	51			STRA, R0	*LNMR	
0176	15F4	1B	04				BCTR, UN	DLY	
0177	15F6	0D	95	4D		RST	LODA, R1	*RITEM	RESTORE ORG VALUE TO R1
0178	15F9	17					RETC, UN		
0179	15FA					DLY	EORI, R0	FF	
0180	15FC	C1					STRA, R1		
0181	15FC	C0					NOP		
0182	15FD	C0					NOP		
0183	15FE	F9	7C				BDRR, R1	\$-2	
0184	1600	F8	7C				BDRR, R0	\$-2	
0185	1602	0D	95	4D			LODA, R1	*RITEM	
0186	1605	77					RETC, UN		
0187	1606	C0				TABLE	DATA	C0	NULL CHARACTER
0188	1607	C0					DATA	C0	ST MSG
0189	1608	C0					DATA	C0	ST TXT
0190	1609	BC					DATA	BC	END TXT
0191	160A	C0					DATA	C0	ENDMT
0192	160B	C0					DATA	C0	WHO RU
0193	160C	C0					DATA	C0	ACK
0194	160D	C0					DATA	C0	BELL
0195	160E	9D					DATA	9D	BACKSP
0196	160F	AF					DATA	AF	HTAB
0197	1610	AE					DATA	AE	LINEFEED
0198	1611	C0					DATA	C0	VTAB
0199	1612	C0					DATA	C0	FORMFEED
0200	1613	AD					DATA	AD	CARRAGE RET
0201	1614	C0					DATA	C0	SHF OUT
0202	1615	C0					DATA	C0	SHF IN
0203	1616	C0					DATA	C0	DL ESC
0204	1617	C0					DATA	C0	X ON
0205	1618	C0					DATA	C0	TAPE ON
0206	1619	C0					DATA	C0	X OFF
0207	161A	C0					DATA	C0	TAPE OFF
0208	161B	81					DATA	81	NO ACK
0209	161C	C0					DATA	C0	SYNCD
0210	161D	9E					DATA	9E	END BLK
0211	161E	C0					DATA	C0	CANCEL
0212	161F	C0					DATA	C0	END MDM
0213	1620	C0					DATA	C0	SUBST
0214	1621	C0					DATA	C0	ESCAPE
0215	1622	C0					DATA	C0	FORM SEP
0216	1623	C0					DATA	C0	GROUP SEP
0217	1624	C0					DATA	C0	RECORD SEP
0218	1625	C0					DATA	C0	UNIT SEP
0219	1626	80					DATA	80	SPACE
0220	1627	01					DATA	01	I

FILE	'TREN'	AS ASSEMBLED BY SYSTEM ON	03-05-79									
LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND		COMMENTS		
0221	1628	49					DATA	49	"			
0222	1629	70					DATA	70	#			
0223	162A	44					DATA	44	\$			
0224	162B	48					DATA	48	%			
0225	162C	68					DATA	68	&			
0226	162D	09					DATA	09	'			
0227	162E	74					DATA	74	(
0228	162F	64					DATA	64)			
0229	1630	78					DATA	78	*			
0230	1631	53					DATA	53	+			
0231	1632	BE					DATA	BE	,			
0232	1633	37					DATA	37	-			
0233	1634	91					DATA	91	.			
0234	1635	07					DATA	07	/			
0235	1636	24					DATA	24	0			
0236	1637	20					DATA	20	1			
0237	1638	10					DATA	10	2			
0238	1639	39					DATA	30	3			
0239	163A	04					DATA	04	4			
0240	163B	08					DATA	08	5			
0241	163C	18					DATA	18	6			
0242	163D	28					DATA	28	7			
0243	163E	38					DATA	38	8			
0244	163F	34					DATA	34	9			
0245	1640	68					DATA	6B	:			
0246	1641	2B					DATA	2B	;			
0247	1642	50					DATA	50				
0248	1643	13					DATA	13	=			
0249	1644	41					DATA	41				
0250	1645	47					DATA	47	?			
0251	1646	50					DATA	50	!			
0252	1647	79					DATA	79	A			
0253	1648	76					DATA	76	B			
0254	1649	7A					DATA	7A	C			
0255	164A	6A					DATA	6A	D			
0256	164B	4A					DATA	4A	E			
0257	164C	73					DATA	73	F			
0258	164D	63					DATA	63	G			
0259	164E	66					DATA	66	H			
0260	164F	59					DATA	59	I			
0261	1650	43					DATA	43	J			
0262	1651	5A					DATA	5A	K			
0263	1652	46					DATA	46	L			
0264	1653	81					DATA	61	M			
0265	1654	52					DATA	52	N			
0266	1655	45					DATA	45	O			
0267	1656	4B					DATA	4B	P			
0268	1657	5B					DATA	5B	Q			
0269	1658	59					DATA	69	R			
0270	1659	65					DATA	65	S			
0271	165A	42					DATA	42	T			
0272	165B	72					DATA	72	U			
0273	165C	71					DATA	71	V			
0274	165D	75					DATA	75	W			
0275	165E	62					DATA	62	X			

FILE	'TREN'	AS ASSEMBLED BY SYSTEM ON	03-05-79									
LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND		COMMENTS		
0276	165F	67					DATA	67	Y			
0277	1660	54					DATA	54	Z			
0278	1661	74					DATA	74				
0279	1662	91					DATA	91				
0280	1663	81					DATA	61				
0281	1664	58					DATA	58				
0282	1665	77					DATA	77	-			
0283	1666	09					DATA	09				
0284	1667	39					DATA	39	A			
0285	1668	36					DATA	36	B			
0286	1669	3A					DATA	3A	C			
0287	166A	2A					DATA	2A	D			
0288	166B	0A					DATA	0A	E			
0289	166C	33					DATA	33	F			
0290	166D	23					DATA	23	G			
0291	166E	26					DATA	26	H			
0292	166F	19					DATA	19	I			
0293	1670	03					DATA	03	J			
0294	1671	1A					DATA	1A	K			
0295	1672	06					DATA	06	L			
0296	1673	21					DATA	21	M			
0297	1674	12					DATA	12	N			
0298	1675	05					DATA	05	D			
0299	1676	0B					DATA	0B	P			
0300	1677	1B					DATA	1B	Q			
0301	1678	29					DATA	29	R			
0302	1679	25					DATA	25	S			
0303	167A	02					DATA	02	T			
0304	167B	32					DATA	32	U			
0305	167C	31					DATA	31	V			
0306	167D	35					DATA	35	W			
0307	167E	22					DATA	22	X			
0308	167F	27					DATA	27	Y			
0309	1680	14					DATA	14	Z			
0310	1681	74					DATA	74				
0311	1682	91					DATA	91				
0312	1683	81					DATA	64				
0313	1684	37					DATA	37				
0314	1685	BF					DATA	BF				

LOWER CASE
CHARACTERS

RUB OUT

Program Mods by Central Data

On the next page and half are the modifications required to bring our most recent software products up to date.

FILE 'ALPMCD' AS ASSEMBLED BY SYSTEM ON 01-17-79

LINE	ADDR	E1	E2	E3	E4	LABEL	OPCODE	OPERAND	COMMENTS
0001	0000					*	ALP	VERSION 1.2 MODIFICATIONS	
0002	0000					*			
0003	0000					*			
0004	0000					*	NEW	LABLE EQUATES	
0005	0000					*			
0006	0000					*			
0007	0000					LASTRM	EQU	496E	
0008	0000					NEWK	EQU	2D9E	
0009	0000					NEWB	EQU	2D8C	
0010	0000					BIN0	EQU	3453	
0011	0000					BITS3	EQU	34A2	
0012	0000					TB1	EQU	3303	
0013	0000					INER0	EQU	36FF	
0014	0000					ITOU0	EQU	478F	
0015	0000					DS2	EQU	3FAC	
0016	0000					DS3	EQU	3F7D	
0017	0000					*			
0018	0000					*			
0019	0000					*	PROGRAM	PATCHES	
0020	0000					*			
0021	0000					*			
0022	496B					ORC	LASTRM		
0023	496B 04 00					PCH1	LODI,R0		
0024	496D FC						WRTD,R0		
0025	496E 17						RETC,UN		
0026	496F 3B 7A					PCH2	BSTR,UN	PCH1	
0027	4971 1F 21 3E						BCTR,UN	DISP	
0028	4974 3B 7F					PCH3	BSTR,UN	PCH1	
0029	4976 07 10						LODI,R3	10	
0030	497E 1F 45 01						BCTA,UN	ERROR	
0031	497B 3B 6E					PCH4	BSTR,UN	PCH1	
0032	497D 1F 2E 0A						BCTA,UN	AOK	
0033	4980 04 24					PCH5	LODI,R0	24	
0034	4982 05 FE						LODI,R1	FE	
0035	4984 06 E2						LODI,R2	E2	
0036	4986 08 7E					PCH6	BRRR,R0	PCH6	
0037	4988 09 7C						BRRR,R1	PCH6	
0038	498A 0A 7A						BRRR,R2	PCH6	
0039	498C 17						RETC,UN		
0040	498D 04 08					PCH7	LODI,R0	08	
0041	49E9 F0						WRTD,R0		
0042	4990 17						RETC,UN		
0043	4991 3B 7A					PCH8	BSTR,UN	PCH7	
0044	4993 1F 2E 0A						BCTA,UN	AOK	
0045	4996 3F 79					PCH9	BSTR,UN	PCH8	
0046	4998 1B 66						BCTR,UN	PCH5	
0047	499A 3E 71					PCHA	BSTR,UN	PCH7	
0048	499C 1F 45 8B					PCHB	BCTA,UN	WRTBL	
0049	499F 3B 79					PCRC	BSTR,UN	PCHA	
0050	49A1 1F 22 59						BCTA,UN	BEGIN	
0051	49A4 3E 74					FCHD	BSTR,UN	PCHA	
0052	49A6 3B 58						BSTR,UN	PCH5	
0053	49AB 1F 34 E6						BCTA,UN	BINADD	
0054	49AB F4 1F					PCHF	TMTI,R0	1F	
0055	49AD 9C 2E 57						BCFA,EQ	BRRMD	

FILE BASICMOD' AS ASSEMBLED BY SYSTEM ON 01-01-79

LINE	ADDR	E1	E2	E3	E4	LABEL	OPCODE	OPERAND	COMMENTS
0001	0000					*			
0002	0000					*	BASIC12	VFRSION 1.2 MODIFICATIONS	
0003	0000					*			
0004	0000					*			
0005	0000					*	PROGRAM	PATCHES	
0006	0000					*			
0007	0000					*			
0008	3FE2						ORC	3FE2	
0009	3FE2 3F 30 66					NEWC3	BSTA,UN	COMK3	
0010	3FE5 1F 04						BCTR,UN	NEWC4	
0011	3FE7 3F 30 66					NEWC3A	BSTA,UN	COMN3	
0012	3FEA 02					NEWC3B	LOE2,R2		
0013	3FE8 1F 30 B5					NEWC4	BCTA,UN	INCSF2	
0014	3FE1 9B 72					NEWC5	PCFR,FC	NEWC3	
0015	3FE0 C2						STRZ,R2		
0016	3FF1 14						RETC,EQ		
0017	3FF2 1F 40 19						BCTA,UN	COMST2	
0018	3FFF 5F 40 19					NEWC6	BRNA,R3	COMST2	
0019	3FF8 1B 70						BCTR,UN	NEWC3B	
0020	3FFA					*			
0021	3FFA					*			
0022	3FFA					*	BASIC12	CHANGES	
0023	3FFA					*			
0024	3FFA					*			
0025	4012						ORC	4012	
0026	4012 05 FF						LOEI,R1	FF	
0027	4014 03						LOE2,R3		
0028	4015 E2						COM2,R2		
0029	4016 1F 3F EF						BCTA,UN	NEWC5	
0030	4019					*			
0031	401F						ORC	401F	
0032	401F 9C 3F E7						BCTA,EO	NEWC3A	
0033	4022 1F 3F F5						BCTA,UN	NEWC6	
0034	4025					*			
0035	4025					*			
0036	4025					*			
0037	3FE7 A7 02						SUBI,R3	02	
0038	3FE9 9A 02						BCTR,LT	DS1	
0039	3FE2 07 0E						LODI,R3	0E	
0040	3FE1 0F 7A D8					DS1	LODA,R3	STACK,I	
0041	3F90 3F 47 4F						BSTA,UN	HXOT	
0042	3F93 2F 7A D9						LODA,R3	STACK+1.I	
0043	3F96 3F 47 4F						BSTA,UN	HXOT	
0044	3F99 E6 01						COMI,R2	01	
0045	3F9B 1C 42 90						BCTA,EO	DCOMD	
0046	3F9E 04 2C						LODI,E0		
0047	3FA0 1B 0A						BCTR,UN	DS2	
0048	3FAC						ORC	DS2	
0049	3FAC 3F 45 8D						BSTA,UN	WRT	
0050	3FAF 3F 45 8E						BSTA,UN	WRTBL	
0051	3FE2 FA 53						BRRR,F2	DS2	
0052	3F7D						ORC	DS3	
0053	3F7D E7 01						ADDI,R3	01	

FILE 'BUGMOD' AS ASSEMBLED BY SYSTEM ON 12-21-78

LINE	ADDR	B1	B2	B3	B4	LABEL	OPCODE	OPERAND	COMMENTS
0001	0000	*					DEBUG VERSION 1.2 MODIFICATIONS		
0002	0020	*							
0003	0000	*							
0004	0000	*					NEW LABEL EQUATES		
0005	0000	*							
0006	0000	*							
0007	0000	DS3					EQU	771A	
0008	0020	*							
0009	0020	*							
0010	0000	*					PROGRAM PATCHES		
0011	0000	*							
0012	0000	*							
0013	7FE0						ORG	FNEPRG	
0014	7FD6	3F	7C	2D		DS2	BSTA,UN	WRT	
0015	7FD3	3F	7C	2F			BSTA,UN	WRTEL	
0016	7FE5	FF	77	24			EDRA,R2	DS2	
0017	7FD9								
0018	7FD9								
0019	7FD5						DEBUG CHANGES		
0020	7FD9								
0021	7FD9								
0022	7724						ORG	DS0	
0023	7724	A7	02				SUBI,R3	02	
0024	7726	9A	02				BCFR,LT	DS1	
0025	7728	07	0E				LODI,R3	0F	
0026	772A	0F	72	75		DS1	LDA,R3	STACK,1	
0027	772D	3F	71	B4			BSTA,UN	EXCT	
0028	7730	0F	72	76			LDA,R3	STACK+1,1	
0029	7733	3F	7D	B4			BSTA,UN	HYOT	
0030	7736	16	01				COMI,R2	01	
0031	7738	1C	79	C4			BSTA,FC	COML	
0032	773B	04	2C				LODI,P0		
0033	773D	1F	7F	D0			BSTA,UN	DS2	
0034	771A						ORG	DS3	
0035	771A	87	01				ADDI,P3	01	
0036	1510								BLKS
0037	1510								ORG
0038	1510	*							
0039	1510	*							
0040	1510	*							
0041	1510	*							
0042	1510 1B 04	DUMP	BCTR,UN	STRT					SKIP OVER POINTERS
0043	1512	*							
0044	1512 00	PTR1	RES	1					
0045	1513 00	PTR2	RES	1					
0046	1514 20 03	STRTXT	ACQN	2003					TELETYPE 43 TEXT PRINTER (300 BAUD)
0047	1516	*							
0048	1516 75 08	STRT	CPSL,	WC					
0049	1518 08 FA	LODR,R0	+STRTXT	GET START OF TEXT AD.					
0050	151A C8 76	STRR,R0	PTR1						
0051	151C 20	EDRZ,R0							
0052	151D C8 74	STRR,R0	PTR2						
0053	151F 3F 00 24	BSTA,UN	LFCR						
0054	1522 3F 15 6D	BSTA,UN	LINF D						
0055	1525 0B EB	LODR,R3	*PTR1	GET CHARACTER FROM TEXT					
0056	1527 1E 15 6B	BCTA,LT	END	MUST BE END OF TEXT					
0057	152A E7 0D	COMI,R3	0D						
0058	152C 38 10	BSTR,EQ	CR						
0059	152E 1A 09	BCTR,LT	OMIT	OMIT NEXT TWO STEPS					
0060	1530 3F 02 4F	BSTA,UN	SERO	OUTPUT CHAR.					
0061	1533 0F 95 12	LDA,R3	*PTR1						
0062	1536 3F 03 96	BSTA,UN	WCHR	WRITE IT ON SCREEN					
0063	1539 3F 15 55	OMIT	BSTA,UN	POINT AT NEXT CHAR.					
0064	153C 18 67	BCTR,UN	GETC	GET ANOTHER ONE					
0065	153E	**							
0066	153E 3F 00 24	CR	BSTA,UN	LFCR	DO CAR.RET ON SCREEN				
0067	1541 07 8A	LDDI,R3	0A	LINEFEED					
0068	1543 3F 02 4F	BSTA,UN	SERO						
0069	1546 07 0D	LODI,R3	0D	CR					
0070	1548 3F 02 4F	BSTA,UN	SERO						
0071	154B 20	EDRZ,R0							
0072	154C 05 30	LODI,R1	30						
0073	154E F8 7E	LOOP	BDRR,R1	LOOP	DO SHALL DELAY				
0074	1550 F9 7C	BDRR,R1	LOOP						
0075	1552 04 FF	LODI,R0	FF	SET CC = LESS THAN					
0076	1554 17	RETC,UN							
0077	1555	*							
0078	1555 0C 15 13	INCPTR	LDA,R0	PTR2					
0079	1558 84 01	ADDI,R0	01						
0080	155A CC 15 13	STRA,R0	PTR2						
0081	155D 15	RETC,GT							
0082	155E 16	RETC,LT							
0083	155F 0C 15 12	LDA,R0	PTR1						
0084	1562 84 01	ADDI,R0	01						
0085	1564 CC 15 12	STRA,R0	PTR1						
0086	1567 17	RETC,UN							
0087	1568	*							
0088	1568 3B 03	END	BSTR,UN	LINF D					
0089	156A 1F 03 C0	BCTA,UN	CMND	RETURN HOME					
0090	156D	*							
0091	156D 3F 15 3E	LINF D	BSTA,UN	CR					
0092	1570 17	RETC,UN							
0093	1571	*							