SPHERE NEWSLETTER

VOLUME V Number 1 September 1980

Editors: Roger J. Spott
Jeffrey Brownstein

SOFTWARE

HARDWARE

EDITOR'S MAILBAG

PLEASE SEND MATERIAL FOR THE NEXT ISSUE TO : DR. ROGER J. SPOTT
13975 Connecticut Avenue
Wheaton, Maryland 20906

ALPHABETICAL SORTING ROUTINE
INSTRUCTIONS

This routine must be loaded into ram from Hex 1000-119F. It starts at 1000 and displays a menu for input,list or order. When you choose input(1) it shows no response, but is ready to accept names, last name first comma first initial period then second initial period. The input area starts at Hex 1200 to 3000. The author of this routine is unknown, and they left no way to correct for spelling errors. The escape convention is the *, which brings you back to the menu. The sort is fast because of machine language and the routine displays the last memory location used in Hex. It might be interesting to play with it or use some of the concepts in your own routine.

Roger

```
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   -----ASCII------

1000 CE 10 36 A6 00 8D 1F 08 8C 10 66 26 F6 8D 1F 81   ..6.......f&....
1010 31 27 0A 81 32 27 09 81 33 27 08 20 E3 7E 10 68   1´..2´..3´.....h
1020 7E 10 F6 7E 11 4B DF C0 BD FC BC DE C0 39 DF C0   .....K.......9..
1030 BD FC 4A DE C0 39 41 4C 50 48 41 42 45 54 49 43   ..J..9ALPHABETIC
1040 41 4C 0D 28 4E 41 4D 45 2C 49 2E 49 2E 29 0D 49   AL.(NAME,I.I.).I
1050 4E 50 55 54 3D 31 20 4C 49 53 54 3D 32 20 4F 52   NPUT=1.LIST=2.OR
1060 44 45 52 3D 33 0D DF 08 86 00 B7 12 00 CE 12 00   DER=3...........
1070 6D 00 27 0A C6 10 08 5A 26 FC 8D 66 20 F2 C6 0E   m.´....Z&..f....
1080 8D 66 81 2A 26 05 6F 00 7E 10 00 81 0D 27 EF 81   .f.*&.o......´..
1090 2E 27 EB 81 2C 27 E7 A7 00 08 5A 27 20 8D 49 81   .´..,´....Z´..I.
10A0 0D 27 06 81 2C 27 10 20 EE 5C 5C 6F 00 08 5A 27   .´..,´...\\o..Z´
10B0 FA 8D 2F 6F 00 20 B9 6F 00 08 5A 26 FA 8D 29 81   ../o...o..Z&..).
10C0 2C 27 FA 81 0D 26 06 4F A7 00 08 20 0E A7 00 08   ,´...&.O........
10D0 8D 16 81 2E 27 FA 81 0D 26 01 4F A7 00 08 8D 02   ....´...&.O.....
10E0 20 9C 8C 30 00 27 0C 39 DF C0 BD FC 4A BD FC BC   ...0.´.9....J...
10F0 DE C0 39 BD FE EC BD FC 37 BD FC 3D CE 12 00 C6   ..9.....7..=....

ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   -----ASCII------

1100 10 6D 00 27 08 A6 00 8D 34 08 5A 20 F4 C1 10 27   .m.´....4.Z....´
1110 34 8C 30 00 27 2F 86 2C 8D 23 08 5A 6D 00 27 FA   4.0.´/.,.#.Zm.´.
1120 A6 00 8D 19 86 2E 8D 15 08 5A A6 00 8D 0F 86 2E   .........Z......
1130 8D 0B 08 6D 00 27 0E 86 0D 8D 02 20 C2 DF C0 BD   ...m.´..........
1140 FC BC DE C0 39 BD FE EC 7E 10 00 CE 12 00 8C 30   ....9..........0
1150 00 27 06 C6 10 6D 10 26 03 7E 10 00 DF D0 A6 00   .´...m.&.......
1160 A1 10 26 06 08 5A 26 F6 20 E4 23 21 C6 10 DE D0   ..&..Z&...#!..
1170 A6 00 97 D3 A6 10 A7 00 96 D3 A7 10 08 5A 26 F0   ..............Z&.
1180 C6 20 09 5A 26 FC 8C 12 00 2D C0 20 C1 08 5A 26   ...Z&....-....Z&
1190 FC 20 BB 84 0F 8B 30 36 BD 01 F1 32 81 20 24 01   ......06...2..$.
```

```
0001                       NAM   SPOTTGEN
0002                       ORG   $1000
0003                 E1    EQU   $00C0
0004                 E2    EQU   $FCBC
0005                 E3    EQU   $FC4A
0006                 E4    EQU   $0008
0007                 E5    EQU   $1200
0008                 E6    EQU   $3000
0009                 E7    EQU   $FEEC
0010                 E8    EQU   $FC37
0011                 E9    EQU   $FC3D
0012                 EA    EQU   $00D0
0013                 EB    EQU   $00D3
0014 1000 CE 1036 L8       LDX   =D1
0015 1003 A6 00   L3       LDAA  0,X
0016 1005 8D 1F            BSR   L1
0017 1007 08               INX
0018 1008 8C 1066          CPX   =L2
0019 100B 26 F6            BNE   L3
0020 100D 8D 1F            BSR   L4
0021 100F 81 31            CMPA  ='1
0022 1011 27 0A            BEQ   L5
0023 1013 81 32            CMPA  ='2
0024 1015 27 09            BEQ   L6
0025 1017 81 33            CMPA  ='3
0026 1019 27 08            BEQ   L7
0027 101B 20 E3            BRA   L8
0028 101D 7E 1068 L5       JMP   L9
0029 1020 7E 10F6 L6       JMP   LA
0030 1023 7E 114B L7       JMP   LB
0031 1026 DF C0   L1       STX   E1
0032 1028 BD FCBC          JSR   E2
0033 102B DE C0            LDX   E1
0034 102D 39               RTS
0035 102E DF C0   L4       STX   E1
0036 1030 BD FC4A          JSR   E3
0037 1033 DE C0            LDX   E1
0038 1035 39               RTS
0039 1036 41      D1       FCC   'ALPHABETICAL'
     1037 4C
     1038 50
     1039 48
     103A 41
     103B 42
     103C 45
     103D 54
     103E 49
     103F 43
     1040 41
     1041 4C
0040 1042 0D               FCB   $D
0041 1043 28               FCC   '(NAME,I.I.)'
     1044 4E
     1045 41
     1046 4D
     1047 45
     1048 2C
     1049 49
     104A 2E
     104B 49
     104C 2E
     104D 29
```

```
--- PAGE 0002
0042 104E 0D             FCB   $D
0043 104F 49             FCC   'INPUT=1 LIST=2 '
     1050 4E
     1051 50
     1052 55
     1053 54
     1054 3D
     1055 31
     1056 20
     1057 4C
     1058 49
     1059 53
     105A 54
     105B 3D
     105C 32
     105D 20
0044 105E 4F             FCC   'ORDER=3'
     105F 52
     1060 44
     1061 45
     1062 52
     1063 3D
     1064 33
0045 1065 0D             FCB   $D
0046 1066 DF 08   L2     STX   E4
0047 1068 86 00   L9     LDAA  =0
0048 106A B7 1200        STAA  E5
0049 106D CE 1200        LDX   =E5
0050 1070 6D 00   LF     TST   0,X
0051 1072 27 0A          BEQ   LC
0052 1074 C6 10          LDAB  =$10
0053 1076 08      LD     INX
0054 1077 5A             DECB
0055 1078 26 FC          BNE   LD
0056 107A 8D 66          BSR   LE
0057 107C 20 F2          BRA   LF
0058 107E C6 0E   LC     LDAB  =$E
0059 1080 8D 66          BSR   L10
0060 1082 81 2A          CMPA  ='*
0061 1084 26 05          BNE   L11
0062 1086 6F 00          CLR   0,X
0063 1088 7E 1000        JMP   L8
0064 108B 81 0D   L11    CMPA  =$D
0065 108D 27 EF          BEQ   LC
0066 108F 81 2E          CMPA  ='.
0067 1091 27 EB          BEQ   LC
0068 1093 81 2C          CMPA  =',
0069 1095 27 E7          BEQ   LC
0070 1097 A7 00   L15    STAA  0,X
0071 1099 08             INX
0072 109A 5A             DECB
0073 109B 27 20          BEQ   L12
0074 109D 8D 49          BSR   L10
0075 109F 81 0D          CMPA  =$D
0076 10A1 27 06          BEQ   L13
0077 10A3 81 2C          CMPA  =',
0078 10A5 27 10          BEQ   L14
0079 10A7 20 EE          BRA   L15
0080 10A9 5C      L13    INCB
0081 10AA 5C             INCB
0082 10AB 6F 00   L16    CLR   0,X
0083 10AD 08             INX
```

```
0084 10AE 5A            DECB
0085 10AF 27 FA         BEQ   L16
0086 10B1 8D 2F         BSR   LE
0087 10B3 6F 00         CLR   0,X
0088 10B5 20 B9         BRA   LF
0089 10B7 6F 00   L14   CLR   0,X
0090 10B9 08            INX
0091 10BA 5A            DECB
0092 10BB 26 FA         BNE   L14
0093 10BD 8D 29   L12   BSR   L10
0094 10BF 81 2C         CMPA  =',
0095 10C1 27 FA         BEQ   L12
0096 10C3 81 0D         CMPA  =$D
0097 10C5 26 06         BNE   L17
0098 10C7 4F            CLRA
0099 10C8 A7 00         STAA  0,X
0100 10CA 08            INX
0101 10CB 20 0E         BRA   L18
0102 10CD A7 00   L17   STAA  0,X
0103 10CF 08            INX
0104 10D0 8D 16   L19   BSR   L10
0105 10D2 81 2E         CMPA  ='.
0106 10D4 27 FA         BEQ   L19
0107 10D6 81 0D         CMPA  =$D
0108 10D8 26 01         BNE   L18
0109 10DA 4F            CLRA
0110 10DB A7 00   L18   STAA  0,X
0111 10DD 08            INX
0112 10DE 8D 02         BSR   LE
0113 10E0 20 9C         BRA   LC
0114 10E2 8C 3000 LE    CPX   =E6
0115 10E5 27 0C         BEQ   L1A
0116 10E7 39            RTS
0117 10E8 DF C0   L10   STX   E1
0118 10EA BD FC4A       JSR   E3
0119 10ED BD FCBC       JSR   E2
0120 10F0 DE C0         LDX   E1
0121 10F2 39            RTS
0122 10F3 BD FEEC L1A   JSR   E7
0123 10F6 BD FC37 LA    JSR   E8
0124 10F9 BD FC3D       JSR   E9
0125 10FC CE 1200       LDX   =E5
0126 10FF C6 10   L20   LDAB  =$10
0127 1101 6D 00   L1D   TST   0,X
0128 1103 27 08         BEQ   L1B
0129 1105 A6 00         LDAA  0,X
0130 1107 8D 34         BSR   L1C
0131 1109 08            INX
0132 110A 5A            DECB
0133 110B 20 F4         BRA   L1D
0134 110D C1 10   L1B   CMPB  =$10
0135 110F 27 34         BEQ   L1E
0136 1111 8C 3000       CPX   =E6
0137 1114 27 2F         BEQ   L1E
0138 1116 86 2C         LDAA  =',
0139 1118 8D 23         BSR   L1C
0140 111A 08      L1F   INX
0141 111B 5A            DECB
0142 111C 6D 00         TST   0,X
0143 111E 27 FA         BEQ   L1F
0144 1120 A6 00         LDAA  0,X
0145 1122 8D 19         BSR   L1C
```

```
0146 1124 86 2E              LDAA  ='.
0147 1126 8D 15              BSR   L1C
0148 1128 08                 INX
0149 1129 5A                 DECB
0150 112A A6 00              LDAA  0,X
0151 112C 8D 0F              BSR   L1C
0152 112E 86 2E              LDAA  ='.
0153 1130 8D 0B              BSR   L1C
0154 1132 08                 INX
0155 1133 6D 00              TST   0,X
0156 1135 27 0E              BEQ   L1E
0157 1137 86 0D              LDAA  =$D
0158 1139 8D 02              BSR   L1C
0159 113B 20 C2              BRA   L20
0160 113D DF C0    L1C       STX   E1
0161 113F BD FCBC            JSR   E2
0162 1142 DE C0              LDX   E1
0163 1144 39                 RTS
0164 1145 BD FEEC  L1E       JSR   E7
0165 1148 7E 1000            JMP   L8
0166 114B CE 1200  LB        LDX   =E5
0167 114E 8C 3000  L25       CPX   =E6
0168 1151 27 06              BEQ   L21
0169 1153 C6 10              LDAB  =$10
0170 1155 6D 10              TST   $10,X
0171 1157 26 03              BNE   L22
0172 1159 7E 1000  L21       JMP   L8
0173 115C DF D0    L22       STX   EA
0174 115E A6 00    L24       LDAA  0,X
0175 1160 A1 10              CMPA  $10,X
0176 1162 26 06              BNE   L23
0177 1164 08                 INX
0178 1165 5A                 DECB
0179 1166 26 F6              BNE   L24
0180 1168 20 E4              BRA   L25
0181 116A 23 21    L23       BLS   L26
0182 116C C6 10              LDAB  =$10
0183 116E DE D0              LDX   EA
0184 1170 A6 00    L27       LDAA  0,X
0185 1172 97 D3              STAA  EB
0186 1174 A6 10              LDAA  $10,X
0187 1176 A7 00              STAA  0,X
0188 1178 96 D3              LDAA  EB
0189 117A A7 10              STAA  $10,X
0190 117C 08                 INX
0191 117D 5A                 DECB
0192 117E 26 F0              BNE   L27
0193 1180 C6 20              LDAB  ='
0194 1182 09       L28       DEX
0195 1183 5A                 DECB
0196 1184 26 FC              BNE   L28
0197 1186 8C 1200            CPX   =E5
0198 1189 2D C0              BLT   LB
0199 118B 20 C1              BRA   L25
0200 118D 08       L26       INX
0201 118E 5A                 DECB
0202 118F 26 FC              BNE   L26
0203 1191 20 BB              BRA   L25
0204                         END
```

BY ROGER J. SPOTT

```
0001 REM THE ITEM IDENTIFICATION NUMBER IS ACTUALLY THE SECTOR NUMBER
0002 REM THIS SOFTWARE KEEPS ALL INFORMATION ON THE DISK. IT PULLS IN THE REQUESTED SECTOR
0003 REM FOR UPDATES AND DUMPS IT BACK TO THE SAME LOCATION.
0004 REM PRINT#5 STATEMENTS SEND COMMANDS TO THE DISK DRIVER
0005 REM PAT STATEMENTS LINES #60 & #68 CHANGE OUTPUT FILE TO #1 AND CHANGE IT BACK TO #2.
0006 REM THE LAST ITEM IS NAMED "END OF FILE"
0010 GOTO 1000
0030 PRINT #3,"ROGER J. SPOTT DDS PA INVENTORY CONTROL ";T$;"          ";F$
0032 PRINT #3,"I.D.#";TAB(10);"ITEM";TAB(25);"ON HAND";TAB(45);"MINIMUM";
0034 PRINT #3,TAB(60);"SUPPLIER CODE";TAB(75);"PRICE/UNIT";TAB(90);
0036 PRINT #3,"LAST ORDERED/AMNT."
0038 RETURN
0040 PRINT #3,D;TAB(6);A$;TAB(25);N;TAB(45);L;TAB(60);S;
0042 PRINT #3,TAB(75);P$;TAB(90);D$
0044 IF A$="END OF FILE" THEN RETURN
0046 RETURN
0060 PAT 8631B7273739
0061 PRINT #5,"P 1 ";D;" 0"
0062 OPEN O"INTORY"
0063 TWRITE D,A$
0064 TWRITE N,L,S
0066 TWRITE P$,D$
0067 CLOSE
0068 PAT 8632B7273739
0069 RETURN
0070 OPEN I "INTORY"
0072 TREAD D,A$
0074 TREAD N,L,S
0076 TREAD P$,D$
0078 RETURN
0200 REM CHANGE DATA
0201 HOME
0202 INPUT "I.D.# TO CHANGE(9999=RETURN)",D
0203 IF D=9999 THEN RETURN
0204 PRINT #5,"P 1 ";D;" 0"
0205 GOSUB 70
0210 HOME
0215 PRINT A$
0220 PRINT D;A$
0225 PRINT N;L;S;P$;" ";D$
0240 PRINT "ITEMS TO CHANGE"
0241 PRINT "1.ITEM NAME"
0242 PRINT "2.ON HAND"
0243 PRINT "3.MINIMUM"
0244 PRINT "4.SUPPLIER CODE"
0245 PRINT "5.PRICE/UNIT"
0246 PRINT "6.LAST ORDER/AMNT"
0247 PRINT "7.DUMP TO DISK"
0250 INPUT "CHOICE",X
0255 ON X GOSUB 510,520,530,540,550,560,60
0260 IF X=5 GOSUB 560
0265 IF X<7 GOTO 210
0290 GOTO 201
0300 REM LIST DATA
0301 REM SET DISK TO SECTOR #0
0305 PRINT #5,"P 1 0 0"
0306 GOSUB 30
0307 PRINT #3
0330 GOSUB 70
0337 GOSUB 40
0338 GOTO 330
```

```
0400 REM ITEMS TO ORDER
0401 REM SET DISK TO SECTOR #0
0405 PRINT #5,"P 1 0 0"
0410 PRINT #3,"ITEMS TO ORDER"
0415 PRINT #3
0420 GOSUB 30
0430 GOSUB 70
0435 IF A$="END OF FILE" THEN RETURN
0440 IF N>=L GOTO 430
0450 GOSUB 40
0460 GOTO 430
0500 INPUT "SECTOR#",D
0501 RETURN
0510 INPUT "ITEM NAME",A$
0511 RETURN
0520 INPUT "ON HAND",N
0521 RETURN
0530 INPUT "MINIMUM",L
0531 RETURN
0540 INPUT "SUPPLIER CODE",S
0541 RETURN
0550 INPUT "PRICE/UNIT",P$
0551 RETURN
0560 INPUT "LAST ORDER/AMNT",D$
0561 RETURN
0600 REM ADD ITEMS
0605 PRINT #5,"P 1"
0610 GOSUB 500:GOSUB 510:GOSUB 520
0615 GOSUB 530:GOSUB 540:GOSUB 550:GOSUB 560
0620 GOSUB 60
0630 INPUT "AGAIN=1",X
0640 IF X=1 GOTO 600
0650 RETURN
```

```
0900 PRINT #3,"SUPPLIER CODE#";TAB(20);"SUPPLIER NAME":PRINT#3
0910 PRINT #3,"0";TAB(20);"UNKNOWN"
0911 PRINT #3,"1";TAB(20);"LOCAL"
0912 PRINT #3,"2";TAB(20);"I.D.E."
0913 PRINT #3,"3";TAB(20);"KAHN PAPER"
0914 PRINT #3,"4";TAB(20);"CODESCO"
0915 PRINT #3,"5";TAB(20);"UNION BROACH"
0916 PRINT #3,"6";TAB(20);"WYETH"
0917 PRINT #3,"7";TAB(20);"SCHEIN"
0918 PRINT #3,"8";TAB(20);"DARBY"
0919 PRINT #3,"9";TAB(20);"ASR MEDICAL"
0920 PRINT #3,"10";TAB(20);"JENSEN"
0921 PRINT #3,"11";TAB(20);"PALMERO"
0922 PRINT #3,"12";TAB(20);"SCHOTT PAPER"
0923 PRINT #3,"13";TAB(20);"MEDIDENTA"
0924 PRINT #3,"14";TAB(20);"PERFECTO"
0925 PRINT #3,"15";TAB(20);"BELVAC"
0926 PRINT #3,"16";TAB(20);"PATTERSON"
0930 RETURN
1000 HOME
1005 INPUT "DATA FILE NAME",X$
1007 INPUT "NEW FILE NAME",F$
1008 PRINT #5,"N ";X$;" ";F$
1009 PRINT #5,"F 1 ";F$
1010 LINE= 132
1015 INPUT "DATE",T$
1020 HOME
1030 PRINT "USE COMMANDS"
1040 PRINT "1.ADD ITEMS"
1050 PRINT "2.CHANGE/CHECK DATA"
1060 PRINT "3.LIST DATA"
1070 PRINT "4.ITEMS TO ORDER"
1090 INPUT "CHOICE",X
1092 IF X>4 GOTO 1020
1093 PRINT #5,"F 1 ";F$
1095 ON X GOSUB 600,200,300,400
1096 PRINT #5,"F "
1097 IF A$="END OF FILE" GOSUB 900
1099 GOTO 1020
```

SAMPLE PARTIAL LISTING OF THE ITEMS IN INVENTORY

ROGER J. SPOTT DDS PA INVENTORY CONTROL SEPTEMBER 9 1980 OFINV.AUG:1

I.D.#	ITEM	ON HAND	MINIMUM	SUPPLIER CODE	PRICE/UNIT	LAST ORDERED/AMNT.
0	ROLL TOWELS	2	3	1	1.09/2	7-80/5
1	PT TOWELS	6	1	16	10.50/CASE	7-80/7
2	NAPKINS	0	0.3	2	10.25/CASE	4-78/1
3	TOILET TISSUE	9	5	3	23.95/CASE	8-77/1
4	FACE TISS	4	1	1	.65/1	7-80/4
5	CUPS	2	2	1	.95/1	3-80/3
6	SOAP	3	2	1	.29/1	3-80/4
7	FILES#08	10	5	16	3.95/PKG	2-80/8
8	FILES#10	100	200	4	.531/1	11-78/450
9	FILES#15	150	100	4	.58/1	12-77/200
10	FILES#20	300	100	4	.58/1	
11	HEDSTROM#15	115	150	16	.575/1	9-78/150
12	HEDSTROM#20	167	150	5	.566/1	1-80/42
13	HEDSTROM#25	175	100	2	.575/1	11-78/48
14	PAPER PTS#15	17	10	7	3.25/BOX	4-80/24
15	PAPER PTS#40	7	10	2	2.60/BOX	7-79/24
16	PAPER PTS#70	7	5	2	2.60.BOX	12-79/6
17	COTT PELLETS#0	1	2	7	1.90/BOX	6-80/1
18	COTT PELLETS#1	2	2	7	1.90/BOX	6-80/1
19	COTT PELLETS#2	2	2	7	1.75/BOX	10-78/1

ITEMS TO ORDER

ROGER J. SPOTT DDS PA INVENTORY CONTROL SEPTEMBER 9 1980 OFINV.AUG:1

I.D.#	ITEM	ON HAND	MINIMUM	SUPPLIER CODE	PRICE/UNIT	LAST ORDERED/AMNT.
0	ROLL TOWELS	2	3	1	1.09/2	7-80/5
2	NAPKINS	0	0.3	2	10.25/CASE	4-78/1
8	FILES#10	100	200	4	.531/1	11-78/450
11	HEDSTROM#15	115	150	16	.575/1	9-78/150
15	PAPER PTS#40	7	10	2	2.60/BOX	7-79/24
17	COTT PELLETS#0	1	2	7	1.90/BOX	6-80/1
21	2 X 2	24	30	16	55.95/CASE	12-79/1
22	GUTTA PERCHA XF	6	10	7	5.50/BOX	6-80/12
25	F.C.	0	1	7	6.35/BOTTLE	2-79/2
32	CAVIT	15	21	7	.99/TUBE	6-30/20
37	FIXER	0	1	1	2.10/ENVELOPE	10-79/4
40	BURS 1557	7	25	16	1.25/1	1-80/12
35	RUBBER DAM	8	10	16	3.35/BOX	5-80/8
64	R.D.CLAMPS#1	2	3	7	2.95/1	6-80/1

```
0001 REM THIS LITTLE PROGRAM IS NICE IF YOU HAVE A LIST OF NUMBERS
0002 REM AND WISH TO RUN A STATISTICAL STUDY OF THEM.
0005 GOTO 1000
0010 INPUT "INPUT NUMBER(9999=END)",X
0020 IF X=9999 RETURN
0030 LET N=N+1
0040 LET S=S+X
0050 LET T=T+X*X
0060 GOTO 10
0200 REM CALCULATIONS
0210 LET M=S/N
0220 LET V=(N*T-S*S)/N/(N-1)
0230 LET D=SQR(V)
0240 LET P=.6745*D
0250 LET E=SQR(V/N)
0260 LET C=D/M
0270 RETURN
0400 REM PRINTER OR SCREEN OUTPUT
0401 HOME
0405 PRINT #Z,"NUMBER ENTIRES",N
0410 PRINT #Z,"TOTAL SUM=",S
0420 PRINT #Z,"SUM-SQUARES=",T
0430 PRINT #Z,"MEAN=",M
0440 PRINT #Z,"VARIANCE=",V
0450 PRINT #Z,"STANDARD DEVIATION=",D
0460 PRINT #Z,"PROBABLE ERROR=",P
0470 PRINT #Z,"STANDARD ERROR OF THE MEAN=",E
0480 RETURN
1000 GOSUB 10
1005 GOSUB 200
1010 INPUT "OUTPUT:SCREEN=1 PRINT=3 STOP=4",Z
1015 IF Z=4 STOP
1020 GOSUB 400
1030 GOTO 1010
```

Roger J. Spott

****

FOR SALE

BOARDS, ALL IN WORKING ORDER

KBD2	$50.00
MEM1	$100.00
CRT1A	$50.00
SIM1	$75.00
PIA1A	$75.00
CPU2	$200.00

TOTAL PACKAGE PRICE $500.00

GREG L. GRINER
259 EARLE DR.
CARLETON MICH. 43117
(313) 654-6873

```
                            NAM       MEM
                            OPT       SYM,LIS        Francis E. Donnelly Jr.

                   * THIS ROUTINE FOR CSS BASIC V4.3 COMPARES MEMEND AND
                   * VARIABLES END, SUBTRACTS THEM, CONVERTS THE RESULT TO
                   * DECIMAL AND OUTPUTS IT AS THE NUMBER OF BYTES LEFT
                   * BETWEEN THE END OF PROGRAM VARIABLE STORAGE AND THE
                   * END OF MEMORY. PUT CMD  MEM  INTO THE COMMAND TABLE
                   * AS   4D 45 4D 00 71 B9

000E               STRHEX   EQU       $000E      TEMP STORE HEX SUBTRACT
0024               AFTVAR   EQU       $0024      FIRST ADR AFTER VARIABLES
0026               MEMLIM   EQU       $0026      MEMORY LIMIT STORED HERE
E07E               PDATA1   EQU       $E07E      MON STRING OUTPUT ROUTINE
E141               CRLF     EQU       $E141      MON CR/LF ROUTINE
E1D1               OUTCH    EQU       $E1D1      MON CHAR OUTPUT ROUTINE

71AD                        ORG       $71AD
71AD 27 10         DECTB1   FDB       10000,1000,100,10
71AF 03 E8
71B1 00 64
71B3 00 0A
71B5               NUMOUT   RMB       1
71B6               TEMP     RMB       1
71B7               TEMP1    RMB       1
71B8               COUNT    RMB       1

                   * THE MAIN PROGRAM STARTS HERE
71B9 96 26         START    LDA A     MEMLIM     GET MSB OF MEMLIM
71BB D6 27                  LDA B     MEMLIM+1   GET LSB OF MEMLIM
71BD D0 25                  SUB B     AFTVAR+1   SUBTRACT LSB'S
71BF 92 24                  SBC A     AFTVAR     SUBTACT MSB OF VAR. END
71C1 97 0E                  STA A     STRHEX     STORE HEX SUB MSB
71C3 D7 0F                  STA B     STRHEX+1   STORE HEX SUB LSB
71C5 BD E1 41               JSR       CRLF       OUTPUT A CR & LF

                   * THE HEX NUMBER IS CONVERTED TO DECIMAL HERE
71C8 7F 71 B5      DECOUT   CLR       NUMOUT     CLEAR NUMBER FLAG
71CB 7F 71 B7      CLRCNT   CLR       TEMP1      SET SUPR 0'S FLAG
71CE 86 04                  LDA A     #$4        LOAD COUNTER VALUE
71D0 B7 71 B6               STA A     TEMP       SET COUNTER
71D3 96 0E                  LDA A     STRHEX     GET MSB
71D5 D6 0F                  LDA B     STRHEX+1   GET LSB
71D7 CE 71 AD               LDX      #DECTB1     POINT TO DECIMAL CONSTANTS
71DA 8D 11         OUTDEC   BSR       OUTDIG     OUTPUT DECIMAL DIGIT
71DC 08                     INX                  BUMP TO NEXT DEC CONSTANT
71DD 08                     INX
71DE 7A 71 B6               DEC       TEMP       DEC THE COUNTER
71E1 26 F7                  BNE       OUTDEC     CONTINUE TILL DONE
71E3 17                     TBA                  GET LS DIGIT
71E4 8D 3A                  BSR       OUTHR      OUTPUT IT
71E6 CE 72 27               LDX       #MEMMSG    POINT TO MESSAGE
71E9 BD E0 7E               JSR       PDATA1     OUTPUT MESSAGE
71EC 39                     RTS                  RETURN TO BASIC
```

```
                       * OUTPUT DECIMAL DIGITS HERE
71ED 7F 71 B8  OUTDIG    CLR       COUNT     CLEAR THE COUNTER
71F0 A1 00     OUTDI2    CMP A     0,X       CHECK THE MSB
71F2 25 0F               BCS       OUTDI5
71F4 22 04               BHI       OUTDI4
71F6 E1 01               CMP B     1,X       CHECK THE LSB
71F8 25 09               BCS       OUTDI5
71FA E0 01     OUTDI4    SUB B     1,X       SUBTRACT LSB
71FC A2 00               SBC A     0,X       SUBTRACT MSB
71FE 7C 71 B8            INC       COUNT     INCR THE COUNTER
7201 20 ED               BRA       OUTDI2    REPEAT TILL < CONSTANT
7203 36        OUTDI5    PSH A               SAVE A
7204 B6 71 B8            LDA A     COUNT     LOAD THE TOTAL
7207 26 10               BNE       OUTDI6    IS IT ZERO YET?
7209 7D 71 B5            TST       NUMOUT    SUPRESS ZERO'S?
720C 26 0B               BNE       OUTDI6    NUMBER YET?
720E 7D 71 B7            TST       TEMP1     NULL?
7211 27 0B               BEQ       OUTDI8
7213 86 20               LDA A     #$20      LOAD A SPACE
7215 8D 0D               BSR       OUTHR2    OUTPUT IT
7217 20 05               BRA       OUTDI8
7219 7C 71 B5  OUTDI6    INC       NUMOUT    BUMP NUMBER
721C 8D 02               BSR       OUTHR     OUTPUT DIGIT
721E 32        OUTDI8    PUL A               RESTORE A
721F 39                  RTS

               * NUMBER OUTPUT ROUTINE
7220 84 0F     OUTHR     AND A     #$F       MASK THE MSB
7222 8B 30               ADD A     #$30      ADD BIAS
7224 7E E1 D1  OUTHR2    JMP       OUTCH     OUTPUT THE DIGIT
7227 20        MEMMSG    FCC       ' BYTES LEFT '
7228 42 59
722A 54 45
722C 53 20
722E 4C 45
7230 46 54
7232 20
7233 04                  FCB       4
                         END
      NO ERROR(S) DETECTED


   SYMBOL TABLE:
AFTVAR  0024      CLRCNT  71CB      COUNT   71B8      CRLF    E141
DECOUT  71C8      DECTB1  71AD      MEMLIM  0026      MEMMSG  7227
NUMOUT  71B5      OUTCH   E1D1      OUTDEC  71DA      OUTDI2  71F0
OUTDI4  71FA      OUTDI5  7203      OUTDI6  7219      OUTDI8  721E
OUTDIG  71ED      OUTHR   7220      OUTHR2  7224      PDATA1  E07E
START   71B9      STRHEX  000E      TEMP    71B6      TEMP1   71B7
```

```
                          NAM     PLIST
                          OPT     PAG,LIS      Francis E. Donnelly Jr.

                    * THIS ROUTINE FOR CSS BASIC V4.3 PROVIDES PAGE LIST
                    * FUNCTION. CURRENTLY SET FOR 59 LINES IN AN 11 INCH
                    * PAGE.

0006                PLFLAG  EQU     $0006   PLIST FLAG: 00=OFF, 01=ON
0007                LINCNT  EQU     $0007   LINE COUNTER TEMPORARY
0008                XTEMP   EQU     $0008   TEMP X REG STORAGE
000A                ATEMP   EQU     $000A   TEMP A REG STORAGE
000B                BTEMP   EQU     $000B   TEMP B REG STORAGE
01F1                OUT     EQU     $01F1   CSS BASIC OUTPUT ROUTINE
02BC                CRLF2   EQU     $02BC   ENTER PDATA1 ROUTINE
0313                LODIDX  EQU     $0313   LOAD X FROM INDEX REG STACK
0BA1                LIST    EQU     $0BA1   START OF CSS BASIC LIST

                    * THIS IS CSS BASIC V4.3 CR/LF ROUTINE MODIFIED BY
                    * PLACING A JMP TO PLIST TEST ROUTINE AT ITS END

02D1                        ORG     $02D1

02D1 8D E9          CRLF1   BSR     CRLF2   OUTPUT CRLF STRING
02D3 0D                     FCB     $D,$A,2,0
02D4 0A 02
02D6 00
02D7 7E 01 40               JMP     PLSTST  GO SEE IF PLIST IS ON

                    * THIS ROUTINE REPLACES JUMP TABLE ADDRESSES NOT
                    * USED AND PROVIDES A TEST TO SEE IF PLIST IS ON.
                    * IF IT IS, LINCNT IS TESTED FOR ZERO. IF NOT ZERO,
                    * EXECUTION OF LIST CONTINUES UNTIL LINCNT=00. THEN
                    * A JSR TO THE PLIST ROUTINE IS MADE AND LF'S ARE
                    * OUTPUT TO ADVANCE TO THE TOP OF THE NEXT PAGE. IF
                    * THE PLIST FLAG IS OFF, NORMAL LISTING EXECUTES.

0140                        ORG     $0140

0140 36             PLSTST  PSH A           SAVE A
0141 7D 00 06               TST     PLFLAG  SEE IF PLIST FLAG IS ON
0144 27 0D                  BEQ     NOPE    NO. DO NORMAL LIST.
0146 96 07                  LDA A   LINCNT  YES, GET LINE COUNT VALUE
0148 4A                     DEC A           DECREMENT IT
0149 97 07                  STA A   LINCNT  SAVE NEW LINE COUNT
014B 26 06                  BNE     NOPE    BRANCH IF LINCNT NOT ZERO
014D 32                     PUL A           LINCNT ZERO. RESTORE A
014E BD 75 CC               JSR     ADVPAG  AND OUTPUT LF'S TO NEXT PAGE
0151 20 01                  BRA     NXTPG
0153 32             NOPE    PUL A           RESTORE A
0154 7E 03 13       NXTPG   JMP     LODIDX  GET X AND CONTINUE LISTING
```

```
                      * THIS IS START OF PLIST ROUTINE LOCATED IN UPPER RAM.
                      * UPON ENTRY THE PLIST FLAG IS SET 'ON', LINCNT IS SET,
                      * AND A HEADER OF 4 LF'S IS OUTPUT. A JSR TO CSS BASIC
                      * LIST ROUTINE IS THEN EXECUTED.

 75AE                           ORG    $75AE

 75AE 86 01           PLIST     LDA A  #$01        LOAD PLIST ON FLAG VALUE
 75B0 97 06                     STA A  PLFLAG      AND SET FLAG
 75B2 86 3C                     LDA A  #$3C        LOAD LINCNT VALUE
 75B4 97 07                     STA A  LINCNT      STORE PAGE LINE COUNT
 75B6 C6 04                     LDA B  #$04        SET FOR 4 LF HEADER
 75B8 8D 2D                     BSR    OUTLF       OUTPUT HEADER LINE FEEDS
 75BA BD 0B A1                  JSR    LIST        GO LIST PAGE

                      * THIS IS ENTRY POINT AT BOTTOM OF PAGE OR END OF
                      * LISTING WHICHEVER COMES FIRST. IF END OF PAGE,
                      * LF'S ARE OUTPUT TO ADVANCE TO TOP OF NEXT PAGE. IF
                      * END OF LISTING, LF'S ARE OUTPUT TO THE BOTTOM OF THE
                      * CURRENT PAGE FOR AN 11 INCH FINAL PAGE LIST.

 75BD 7D 00 07        TSTCNT    TST    LINCNT      SEE IF LINCNT IS ZERO
 75C0 27 0A                     BEQ    ADVPAG      LF TO TOP OF NEXT PAGE
 75C2 8D 15                     BSR    SAVXAB      LIST DONE, NOT BOTTOM OF PAGE
 75C4 D6 07                     LDA B  LINCNT      GET LINE CNT REMAINING
 75C6 8D 1F                     BSR    OUTLF       OUTPUT LF'S TO PAGE BOTTOM
 75C8 7F 00 06                  CLR    PLFLAG      CLEAR PLIST FLAG
 75CB 39                        RTS                ALL DONE. RETURN TO CMD MODE.

                      * OUTPUT LF'S TO TOP OF NEXT PAGE

 75CC 8D 0B           ADVPAG    BSR    SAVXAB      SAVE X,A, & B
 75CE C6 09                     LDA B  #$09        LOAD B WITH 9 LF COUNT
 75D0 8D 15                     BSR    OUTLF       OUTPUT LF'S TO TOP NEXT PAGE
 75D2 86 3C                     LDA A  #$3C        RE-INITIALIZE PAGE LINE COUNT
 75D4 97 07                     STA A  LINCNT      STORE IT IN LINCNT
 75D6 8D 08                     BSR    LODXAB      RESTORE X,A, & B
 75D8 39                        RTS

                      * HERE X,A, & B ARE SAVED WHILE LF'S ARE OUTPUT.

 75D9 DF 08           SAVXAB    STX    XTEMP       TEMP STORE X
 75DB 97 0A                     STA A  ATEMP       TEMP STORE A
 75DD D7 0B                     STA B  BTEMP       TEMP STORE B
 75DF 39                        RTS
```

```
                * HERE X,A, & B ARE RESTORED AFTER LF'S OUTPUT.

 75E0 DE 08     LODXAB  LDX    XTEMP    RESTORE X
 75E2 96 0A             LDA A  ATEMP    RESTORE A
 75E4 D6 0B             LDA B  BTEMP    RESTORE B
 75E6 39                RTS

                * THIS IS THE SUBROUTINE TO OUTPUT LF'S. THE NUMBER OF
                * LF'S IS DETERMINED BY VALUE OF B-REGISTER AND IS
                * EITHER 09 FOR ADVANCE TO NEXT PAGE OR THE VALUE
                * LEFT IN LINCNT ($0007) IF END OF LIST WITH BOTTOM
                * OF PAGE NOT REACHED.

 75E7 86 0A     OUTLF   LDA A  #$0A     LOAD A WITH LF
 75E9 8D 04     MORE    BSR    OUTS     OUTPUT LF
 75EB 5A                DEC B           DECREMENT B LF COUNT
 75EC 26 FB             BNE    MORE     DO TILL B IS ZERO
 75EE 39                RTS

 75EF BD 01 F1  OUTS    JSR    OUT      OUTPUT LINE FEED
 75F2 39                RTS
                        END
      NO ERROR(S) DETECTED


   SYMBOL TABLE:
ADVPAG  75CC      ATEMP   000A      BTEMP   000B      CRLF1   02D1
CRLF2   02BC      LINCNT  0007      LIST    0BA1      LODIDX  0313
LODXAB  75E0      MORE    75E9      NOPE    0153      NXTPG   0154
OUT     01F1      OUTLF   75E7      OUTS    75EF      PLFLAG  0006
PLIST   75AE      PLSTST  0140      SAVXAB  75D9      TSTCNT  75BD
XTEMP   0008
```

Size for CSS BASIC

By Robert Grainger, Jeffrey Brownstein and Frank Donnelly

This routine compares MEMEND and VARIABLES END, subtracts them, converts the result to decimal and places the result on the CRT.

Using the V3N prom routines

```
BD 17EC      Home
CE 000A
DF 04
D6 26
96 27
90 25
D2 24
CE E000      Screen location
7E FF64      Binary to Ascii
```

CSS BASIC NEW COMMAND   TCALL

```
BD 09AA
BD 23D6
DF FA
BD 2D03
BD 17EC
BD 2461
7E 2492
```

This calls another program: Chan Wai Yung

MEDIUM RESOLUTION LIGHT PEN FOR SHERES

The object of this project was to construct a light pen device which would be useful in selecting options from a menu format on the 32 or 64 character CRT1 screen. It was further planned that the hardware modification should be minimal.

I purchased some starting hardware from an add in Byte Magazine. The 3G "APPLE PROFESSIOAL" model light pen costs about thirty dollars and does not even come with a schematic or pin connections. I ha d specifically asked for these things. It was easy to find out the port pin diagram at my local computer store. As for the circuit, there is a photodiode and a two transistor amplifier. The pins(1 ground, 6 output, 8 five volt supply) are easily hooked up to a PIA output socket for initial trials. There is a sensitivity control but even at the most sensitive the CRT glare shield must not be covering the screen. I understand that most light pens have the same limitation.

The light pen respondded easily to a blinking cursor so I set out to devise some software to use it. The simple approach is to start with a dark screen and pass a cursor across it. Presumably when the light is sensed by the pen the index register contains the location on the screen which is lit. In reality this does not work. The pen senses light but by the time that the PIA (either input or CA line) knows it, the software is busy putting cursors far ahead on the screen. Also, if you scan the cursor too fast, it breaks up. If you do n ot scan it fast, the screen takes too long to cover. A practical light pen should read fast.

I tried to speed up the hardware by placing a 74123 one shot after the transistor output. In the retriggerable multivibrator mode, this chip holds the output and creates much longer pulses (almost one continuous pulse). The idea was for the PIA to have a better chance of reading when the pulse was there. This helped some but there is still a time lag to conquer. I placed a loop of sixty reads to make certain that the valid signal was not missed by the PIA. This slows down the scanning of the screen because after every spot is lit, there is the wait to see if it has been sensed.

The solution to this problem was the following: Store the screen full of menu items away in a buffer as a whole screenfull. Only bring back locations with cursor or printed matter onto the screen. This is pretty fast. A relatively small number of locations are actually being returned on the screen and being evaluated by the read loop.

A CSS Basic program is presented to show how one may determine which of three items the light pen pointed at.

```
10 HOME
20 PRINT CHR$(134); "ITEM 1 "
30 SKIP 4
40 PRINT CHR$(134); "ITEM 2 "
50 SKIP 4
60 PRINT CHR$(134); "ITEM 3 "
70 PAT 7FF0427FF04386FFB7F04339      (intitialize PIA)
80 PAT B6F04226FB39                  (look for any light)
90 PAT 7EC000                        (goto PEN routine)
```

Pen routine stores location of the read in FE,FF. Use PEEK to find out if those locations contain the address of the desired item.

```
?  C000 7F CLR  F042
? C003 7F CLR  F043
? C006 86 LDAA#FF
? C008 B7 STAA F043
? C00B B6 LDAA F042
? C00E 26 BNE  FB         C00B
? C010 C6 LDAB#60
? C012 CE LDX #B000
? C015 DF STX  FC
? C017 CE LDX #E000
? C01A DF STX  FE
? C01C A6 LDAAX00
? C01E E7 STABX00
? C020 DE LDX  FC
? C022 A7 STAAX00
? C024 08 INX
? C025 DF STX  FC
? C027 DE LDX  FE
? C029 08 INX
? C02A 8C CPX #E3FF
? C02D 26 BNE  EB         C01A
? C02F CE LDX #E000
? C032 DF STX  FE
? C034 CE LDX #B000
? C037 DF STX  FC
? C039 A6 LDAAX00
? C03B DE LDX  FE
? C03D 81 CMPA#60
? C03F 27 BEQ  0E         C04F
? C041 A7 STAAX00
? C043 C6 LDAB#60
? C045 B6 LDAA F042
? C048 26 BNE  02         C04C
? C04A 20 BRA  0F         C05B
? C04C 5A DECB
? C04D 26 BNE  F6         C045
? C04F 08 INX
? C050 DF STX  FE
? C052 DE LDX  FC
? C054 08 INX
? C055 8C CPX #B3FF
? C058 26 BNE  DD         C037
? C05A 39 RTS
? C05B 09 DEX
? C05C A6 LDAAX00
? C05E 81 CMPA#86
? C060 26 BNE  F9         C05B
? C062 DF STX  FA
? C064 BD JSR  FC37
? C067 BD JSR  FC3D
? C06A 39 RTS
```

PEN

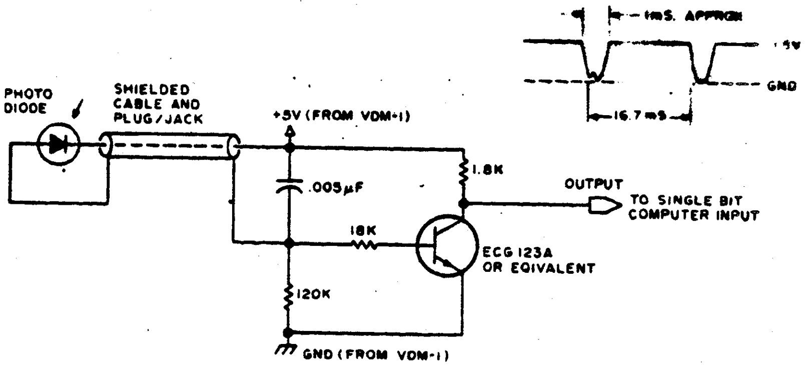
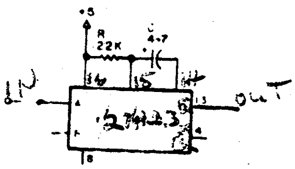


Figure 1: This fairly simple circuit transforms light from the video monitor output into a TTL signal which can be monitored by a program such as that shown in figure 2 and listing 1. A typical oscilloscope waveform of the output is shown in the figure.

74123



STRETCHES OUT PUT Pulses

J.C.Pirtle

ACTIVE TERMINATOR

Bus terminations (specially data bus) will obviously improve the reliability of most Sphere systems. A standard passive termination works very well but the added power dissipation is of course a disadvantage.

An Active Termination should work equally well while minimizing power dissipation.

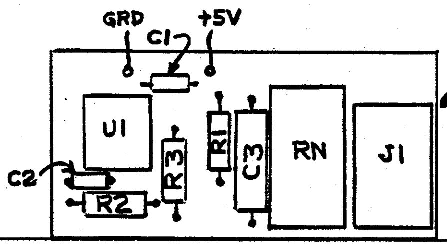
The terminator described below uses an LM13080N IC which is capable of sourcing or sinking 250 mA and operating from a single 5V supply. With the R2/RN values shown the circuit provides a standard 2.6V/180 ohm termination. If you require a different termination, select R2/RN as required:

$$E_t = \frac{Ein}{\left(1 + \frac{R1}{R2}\right)} \qquad R2 = \frac{R1}{\left(\frac{Ein}{E_t} - 1\right)}$$

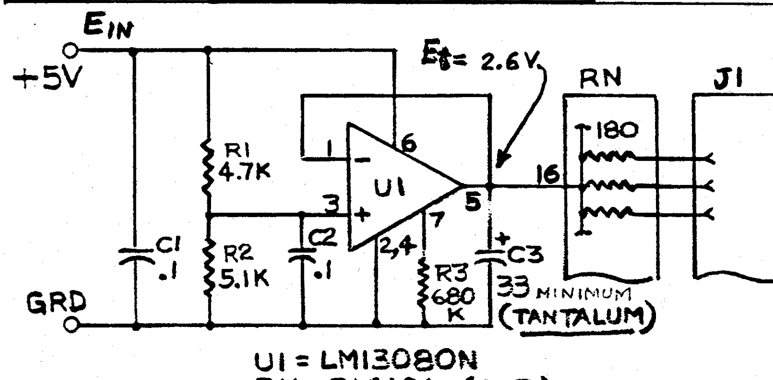
The +5 volt lead need not be large, but the ground lead should be a heavy guage (18 Min.) short wire to your prime ground. A low Z ground lead is a must. A "bonding braid" ground lead may inprove performance in some systems.

REFERENCE:

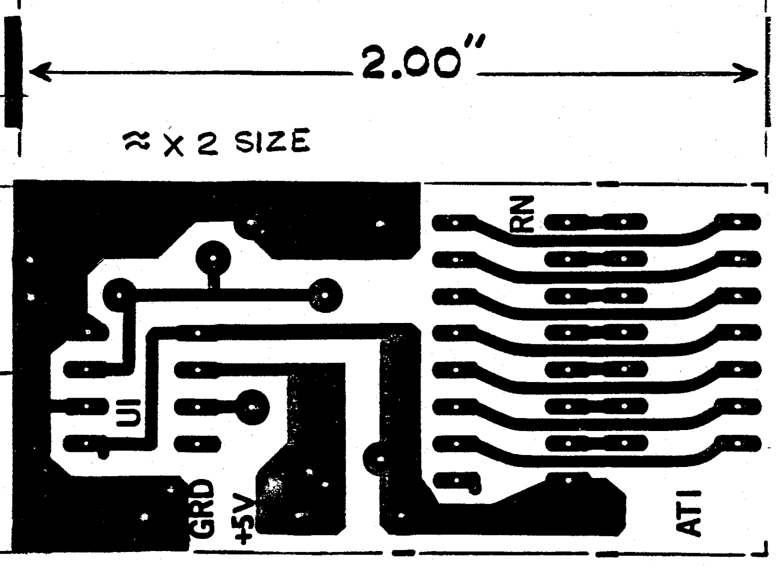
1. HOW IMPORTANT IS PROPER TERMINATION, KILOBAUD, APRIL 1979
2. EXPLORING THE INEQUALITY OF BUS BUFFERS, KILOBAUD, OCT. 1979



ACTUAL SIZE PARTS LAYOUT



ACTIVE TERMINATOR, (2.6V.)



INEXPENSIVE COLOR GRAPHICS (AND ALPHANUMERICS)

There is a game/computer on the market which contains a 6800 CPU as well as 1K of static ram and some rom. This small module also contains a power supply and an entire color TV modulator section. The game attaches to the antenna terminals of the color (or B/W) set. The game is called "Imagination Machine".

Designed to sell as a whole computer, the basic above described section mates with a keyboard/cassette recorder section. One can purchase the small module alone for only $99 and a complete technical manual with schematics is available for a dollar or two from the manufacturer.

What I am proposing is that one can easily hang the 1K static ram dual port memory on any 6800 bus and access the rom graphics routines possibly as well. The internal 6800 would be removed from its socket for this application.

There is clever usage of the 1K ram in that 500 bytes are used to represent screen locations while the other 500 are used to set up various character forms. This makes for medium level resolution but is rather easy to manipulate when doing the programming. Letters and numbers are easily displayed in various colors.

Since the Imagination Machine is intended for department store distribution, it may possibly be available at a discount during the Christmas season. Another possibility is that many of these were sent to computer stores for evaluation without charge. My local store gave its sample back to the rep but they may be able to get it back. These samples sometimes "dis appear " while being evaluated. One dealer tried to return the sample but was told to hold on to it indefinitely until disk drives come out for this machine.

The Imagination Machine does pretty well as a black and white graphics system and presents the colors in various shades of grey. Some day a color TV may come your way with a jammed tuner or no sound. This would certainly reduce the cost. Last week I saw a surplus color monitor for only $125 but it was only a 9 inch screen. For color graphics I feel that a much larger viewing area would be more appropriate.

Jeff

WANTED: a KEYBOARD 2

WANTED: tape of Programma HOME ACCOUNTING program. I still have the instruction sheet but have lost the cassette containing this program.

Jeff

WANTED: Articles for the Newsletter

FDOS 1 DISK MODULE

I have written a module which goes in high memory or Rom and allows any calling program to allocate, delete or examine files on either disk drive. Previously available modules, specifically DISPDISK and EDITOR, took up lots of ram space and could not run at the same time as applications. I have Basic call the module to enable me to allocate files etc. while a Basic program and the interpreter are in memory.

HASHING MODULE

I have written a machine code module which is accessed by Basic and which provides a two byte hashed address from a string of any size. This can be used for random access schemes to provide fast disk sector storage and retrieval.

These program listings are available if anyone can use them.

Jeff

IF YOU HAVE NOT SENT IN YOUR RENEWAL CHECK THIS ISSUE WAS SENT TO YOU FREE AND WILL BE YOUR LAST ISSUE.

2510 Broadway
Big Spring, Texas 79720
September 10, 1980

Dear Roger,

This summer I got Jeff's Sphereized version of the Canadian Pascal running in my machine with only one stumbling block--I need to link it to my printer routine and I don't know where the locations are that get utilized when Pascal sees a WRITE or WRITELN. I need to know also whether Pascal expects to pass characters to the print routine or whether it passes beginning and ending addresses of the string to be printed and lets the print routine get its own characters. I assume it puts a single character in the A register and then does a JSR but I don't know where that JSR is. Perhaps you or one of the Newsletter readers can help me through this problem.

I haven't worked on it in quite a while because I've been getting myself a MECA BETA-1 and am now in the process of making its interface connections to my PIM-5 board. I appreciate the work of Larry Sambuco on this and the help he gave me. When I get it working I'll send a report and a listing of my driver program, if I succeed in getting one different from Larry's. The step after that will be to put it on a PROM. I had thought of just replacing the cassette driver routines but it would be nicer to put a PROM board in the address space like from D000-DFFF and develop a real operating system. Are Sphere PROM boards still available anywhere? If so I would like to know who has one and at what price. An unpopulated one and a set of schematics would be about my speed too!

Thanks for your efforts with the newsletter--I hope it can be kept viable as our Spheres age. You might mention in the next newsletter whether Matteson is still willing to supply spare parts--haven't heard much about him in recent newsletters.

Sincerely,

Joe

Joseph Dawes