

# SPHERE

## NEWSLETTER

AUGUST 1979

VOLUME IV ISSUE I

EDITORS:

ROGER J. SPOTT

JEFF BROWNSTEIN

## SOFTWARE

BASIC

ENERGY SIMULATION	DOUG CALLEY	PAGE 1
POSTER PROGRAM	CHAN WAI YUNG	PAGE 6
SPHEREFORTH PATCHES	T.R. MEIER	PAGE 10
COSMAC ELF - SPHERE MONITOR INTERFACE	DONALD DORSON	PAGE 12
ADD TELETYPE TO TINY BASIC	BILL RUTHERFORD	PAGE 17

## HARDWARE

LOW COST DIGITAL TAPE SYSTEM	J.C. PIRTLE	PAGE 21
------------------------------	-------------	---------

## EDITOR'S MAILBAG

COMPUTERWARE BASIC AT A DISCOUNT	PAGE 27
SCOTT ADAMS GRAPHICS SOFTWARE	PAGE 27
CONVERT CRT-80 SOFTWARE TO MULTIPLE/SPLIT SCREEN EDITOR	PAGE 28

PLEASE SEND TYPED MATERIAL FOR NEXT ISSUE TO:

ROGER J. SPOTT  
13975 CONNECTICUT AVENUE  
WHEATON, MARYLAND 20906

ENERGY SIMULATION, A GAME WHERE YOU SELECT YOUR PREFERENCES FOR ENERGY USE (RICH DIET, CENTRAL HEATING, AMOUNT OF DRIVING, WASHER-DRYER, ETC) HOWMUCH YOU EXPECT TO "SPEND" IN EDUCATION AND IN RESEARCH AND DEVELOPMENT, THE PROGRAM GIVES YOU CUES AS TO THE QUALITY OF YOUR ENVIRONMENT.

JUNE 1979; DOUG CALLEY

NOTE: FOR CLEAR LINES ... SEE END OF PROGRAM

```
10 CLEAR
20 DIM A(11),B(10),C(10),P(10),S(255),T(255),Z$(1)
30 PRINT "          ENERGY"
40 PRINT \ PRINT
50
60 FOR I=1 TO 10
70 LET A(I)=0\B(I)=0\C(I)=0
80 NEXT I
90 LET P(10)=0\E=200\V=1
100 FOR I=1 TO 100
110 LET T(I)=0\S(I)=0
120 NEXT I
125 LET M=0
130 LET Y=0\Q=1\A(5)=100\A(3)=1
140 LET C=680\O=1000\G=620\H=3 N=3.8
150 INPUT Z
160 GOSUB 240
170 GOSUB 650
180 GOSUB 710
190 GOSUB 810
200 GOSUB 920
210 GOSUB 1070
220 GOSUB 1180
230 GOTO 1720
240 CLEAR
250
260 LET P(1)=0\P(3)=0\P(4)=0\P(5)=0
270 IF Z$="Y" P(1)=P(1)+0.75
280 PRINT "DO YOU HAVE A SMALL B/W SET?"\ INPUT Z$
290
300 PRINT "DO YOU OWN OTHER TV SETS?"\ INPUT Z$
310 IF Z$="Y" P(1)=P(1)+0.4
320 PRINT "DO YOU HAVE A LARGE 'STFREQ'?"\ INPUT Z$
330 IF Z$="Y" P(1)=P(1)+0.4
340 PRINT "DO YOU OWN A DISHWASHER?"\ INPUT Z$
350 IF Z$="Y" P(3)=P(3)+0.5
360 PRINT "DO YOU USE A WASHING MACHINE?"\ INPUT Z$
370 IF Z$="Y" P(3)=P(3)+0.3
380 PRINT "DO YOU USE A DRYER?"\ INPUT Z$
390
400 PRINT "HOW MANY ROOMS DOES YOUR HOUSE HAVE?"\ INPUT Z
410 IF Z=0 GOTO 400
420 LET P(7)=Z
430
436 FOR Z=1 TO 1500
437 NEXT Z
440 CLEAR \ PRINT "DO YOU USE AN AIR CONDITIONER?"\ INPUT Z$
450 IF Z$="Y" P(3)=P(3)+3+P(7)/2 \ PRINT "NICE, BUT A HEAVY ENERGY-USER."
460 PRINT "DO YOU HAVE AN EVAPORATIVE          COOLER?"\ INPUT Z$
```

```

470 IF Z$="Y" P(3)=P(3)+0.5+P(7)/3
480 PRINT "DO YOU HAVE ELECTRIC LIGHTS?"\ INPUT Z$
490 IF Z$="Y" P(4)=P(4)+1+P(7)/2
495 IF Z$="N" GOTO 540
500 PRINT "DO YOU HAVE OUTSIDE FLOODLIGHTS?"\ INPUT Z$
510 IF Z$="Y" P(4)=P(4)+5+P(7)/5
720 CLEAR
530
540 PRINT "DO YOU HAVE A REFRIGERATOR?"\ INPUT Z$
550 IF Z$="N" GOTO 590
560 IF Z$="Y" P(5)=P(5)+1
570 PRINT "IS IT 'FROST FREE'?"\ INPUT Z$
580
590 PRINT "DO YOU HAVE A FREIZER?"\ INPUT Z$
600 IF Z$="N" GOTO 640
610 IF Z$="Y" P(5)=P(5)+1
620 PRINT "IS IT FROST-FREE?"\ INPUT Z$
630 IF Z$="Y" P(5)=P(5)+2
640 RETURN
650 LET P(6)=0
660 PRINT "DO YOU HAVE CENTRAL HEATING?"\ INPUT Z$
670 IF Z$="Y" P(6)=P(6)+10+P(7)/2
675 IF Z$="Y" GOTO 700
680
690 IF Z$="Y" P(6)=P(6)-3-P(7)/4
700 RETURN
710 PRINT "DO YOU OWN A CAR?"\ INPUT Z$
715 LET P(8)=0
720 IF Z$="N" PRINT "THAT'S A RELIEF!"\ GOTO 790
730 PRINT "HOW MANY MILES DO YOU DRIVE ONE WAY TO WORK?"\ INPUT Z
740 LET P(8)=10+7*Z/10
750 PRINT "DO YOU RIDE A BUS OR CAR POOL?"\ INPUT Z$
760 IF Z$="Y" P(8)=P(8)-7*Z/20
770
780 LET P(8)=P(8)*(1+(Z-2)/10)
785
790 FOR Z=1 TO 1000
795 NEXT Z
797 LET Z=0
800 RETURN
810 CLEAR
820 PRINT "CONSIDER A BREAKFAST MENU:"
830
840 LET P(9)=0
850 PRINT "TYPE THE # OF THE MENU WHICH BEST SUITS YOU." \ INPUT Z
860 LET P(9)=P(9)+10-2*Z
865 CLEAR
870
880
890 LET P(9)=P(9)+20-4*Z
900 LET Z=0
910 RETURN
920 CLEAR
930
931
940 PRINT \ PRINT "BEFORE, YOU USE": A(1); A(2); M; A(4)
950 PRINT "INTO FERTILIZER" \ INPUT Z
960 LET A(1)=Z
970 PRINT "INTO CULTIVATION?" \ INPUT Z
980 LET A(2)=Z
990 PRINT "INTO R&D" \ INPUT M
1010
1020 LET A(4)=Z
1030 LET Z=A(1)
1040 IF A(2)<A(1) Z=A(2)

```

```

1050 LET Z=0
1060 RETURN
1065
1070 PRINT "HOW MUCH ENERGY DO YOU WANT TO INVEST IN EDUCATION?"\ INPUT
1080 IF N>10L=10\ PRINT "MAX IS 10 EU."
1100 PRINT "HOW MUCH ENERGY DO YOU WANT TO USE IN REFORESTATION?"\ INPUT
1110 IF F>10F=10\ PRINT "(MAX IS 10)"
1120 LET F=F+Z
1130 PRINT "WHAT IS A REALISTIC POPULATION GROWTH ?"\ INPUT K
1160 LET Z=0
1170 RETURN
1180 CLEAR
1190 REM CALCULATE ANOTHER YEAR- NO CHANGES
1200 LET P=0J=0
1205 LET A(1)=A(1)+M
1206 LET A(3)=(A(1)/A(3)+A(3))/2
1207 IF ABS(A(1)/A(3)-A(3))>=0.001 GOTO 1206
1210 LET T(Y+10)=A(3)
1212 LET R=1+K/100-S(Y)/200\ IF A(5)<0R=R+A(5)/200
1213 IF Y<20F=R*1.03
1230 LET P=P(1)+P(2)+P(3)+P(4)+P(5)+P(6)+P(8)
1245 LET S(Y+05)=L
1250 LET Y=Y+1\Q=Q*R
1260 LET C=C-C(1)\O=0-C(2)\G=G-C(3)\H=3+T(Y)\N=3.8+T(Y)/3
1270 IF C<0 PRINT "IMPOSSIBLE"\ GOTO 1980
1280 IF O<0 PRINT "IMPOSSIBLEG"\ GOTO 1980
1290 IF G<0 PRINT "IMPOSSIBLE"\ GOTO 1980
1300 IF H<0 PRINT "IMPOSSIBLE"\ GOTO 1980
1310 IF N<0 PRINT "IMPOSSIBLE"\ GOTO 1980
1320 LET A=A(1)+A(2)+M+A(4)
1330 LET Z=A(1)\ IF A(2)<A(1)Z=A(2)
1340 LET A(5)=A(5)-A(4)/2+T(Y)*2+Z*2-P(9)*Q+6
1345 IF A(5)<0 PRINT "PEOPLE STARVED!"\R=R+A(5)/500
1347 IF A(5)<-5 PRINT \ PRINT "MALNUTRITION AND PESTILENCE!"\ PRINT
1350 LET D=P+A+S+F
1360 LET E=E+C(1)+C(2)+C(3)+C(4)+C(5)-D*1.5*Q
1370
1371 PRINT "SO2 IN AIR IS";B(2)
1380 IF E<0V=V+E/5000\E=0
1390
1400 IF V<0.25 PRINT "THE COUNTRY IS BANKRUPT"\ END
1410 LET B(1)=B(1)+C(1)/(3+0.1*F)- F/20
1620 IF B(1)<0B(1)=0
1430
1440 IF B(1)>70 PRINT \ PRINT "I CAN HARDLY BREATHE"\ PRINT \ PRINT
1450 IF B(1)>50 PRINT "THE AIR IS TERRIBLE"
1660 LET B(2)=B(2)+(C(1)+C(2)+C(3))/(10+0.3*F)- F/10-J
1470 IF B(2)<0B(2)=0
1480
1490
1500 IF B(2)>60 PRINT "THE AIR IS BAD"
1510 LET B(3)=B(3)+C(5)/2+C(1)/20-0.005*B(3)
1720 IF B(3)<0B(3)=0
1530
1540
1550 IF B(3)>50 PRINT "**DANGER OF RADIATION SICKNESS**"\ PRINT
1760 LET Z=0
1670 RETURN
1580 CLEAR
1590 REM CHOOSE AMOUNT TO F COAL,OIL,ETC TO USE TO SUPPLY ENERGY
1600 PRINT "THE AVAILABLE ENERGY IS";
1610 PRINT " COAL OIL GAS NON-POLUT NUCLEAR"
1620 POKE 240,0\ PRINT C;O;G,\ POKE 240,2\ PRINT H;N
1625
1630

```

```

1650 IF C<C(1) PRINT "IMPOSSIBLE"\ GOTO 1600
1660 IF O<C(2) PRINT "IMPOSSIBLE"\ GOTO 1600
1670 IF G<C(3) PRINT "IMPOSSIBLE"\ GOTO 1600
1680 IF H<C(4) PRINT "IMPOSSIBLE"\ GOTO 1600
1690 IF N<C(5) PRINT "IMPOSSIBLE"\ GOTO 1600
1710 RETURN
1720 CLEAR
1730 PRINT "PERSONS. E.DEMAND: "; D: "EL. POP: "; Q
1740 PRINT "PERSONS. FOOD NEEDS: "; P(9)
1745 IF A(5)<0 PRINT "NO FOOD! PEOPLE STARVED" "\Q=Q+A(5)/500
1750 PRINT "THE AVAILABLE ENERGY IS: ";
1760 PRINT "          COAL OIL GAS          HYDRO NUCLEAR"
1770 POKE 240,0\ PRINT C;0;G;\ POKE 240,2\ PRINT H;N
1780 PRINT "USEF: "; C(1);C(2);C(3),C(4),C(5)
1790 PRINT "ENVIRN:PARTIC SO2 RADIATION"
1800 PRINT "          ":B(1)-B(2)-B(3)
1810 PRINT "FOODPOOL: ";A(5);"ENERGYPOOL: ";E
1820 POKE 240,2\ PRINT "THE DOLLAR IS WORTH";V;\ POKE 240,0\ PRINT "YES"
1830 PRINT "CHOOS: "; INPUT Z
1835 IF Z>0 IF Z<9 GOTO 1870
1840
1850
1860 INPUT Z
1870 IF Z=0 GOSUB 1180
1880 IF Z=1 GOSUB 240
1890 IF Z=2 GOSUB 650
1900 IF Z=3 GOSUB 710
1910 IF Z=4 GOSUB 810
1920 IF Z=5 GOSUB 920
1930 IF Z=6 GOSUB 1070
1940 IF Z=7 GOSUB 1720
1950 IF Z=8 GOSUB 1580
1960 GOTO 1830
1970 END
1980 LET C=C+C(1)\O=O+C(2)\G=G+C(3)
1990 GOTO 1600

```

```

50 PRINT "CAN YOU SURVIVE IN THIS
WORLD? IN THIS SIMULATION YOU
ARE ONLY ONE PERSON BUT WE SHALL
ASSUME THAT ALL PEOPLE LIVE
AS YOU DO. WE NEED TO KNOW HOW
YOU WANT TO SPEND YOUR ENERGY."

```

```

250 PRINT "THIS SECTION ASKS ABOUT
HOUSEHOLD USE OF ENERGY. DO YOU
HAVE A CONSOLE COLOR TV?"\ INPUT Z$

```

```

260 LET P(1)=0\ P(3)=0\ P(4)=0\ P(5)=0
270 IF Z$="Y" P(1)=P(1)+0.75
280 PRINT "DO YOU HAVE A SMALL
B/W SET?"\ INPUT Z$
290 IF Z$="Y" P(1)=P(1)+0.2\ IF
P(1)=0.2 PRINT "THAT'S GOOD,
THEY DON'T TAKE MUCH POWER."

```

```

390 IF Z$="Y" P(3)=P(3)+1.4\ PRINT
"DO YOU REALLY NEED ONE? WELL,
KEEP IT FOR NOW."

```

```

400 PRINT "HOW MANY ROOMS DOES
YOUR HOUSE HAVE?"\ INPUT Z

```

```

410 IF Z=0 GOTO 400

```

```

620 LET P(7)=Z

```

```

430 CLEAR \ PRINT "I SHALL ASK
ABOUT AIR CONDITIONERS
AND EVAPORATIVE COOLERS."

```

```

530 IF Z$="Y" PRINT "PLEASE CHECK
WITH YOUR LOCAL OBSERVATORY
ABOUT LIGHT POLLUTION"
\ PRINT \ PRINT \ PRINT

```

```

540 PRINT "DO YOU HAVE A REFRIG
FRATOR?"\ INPUT Z$

```

```

550 IF Z$>"N" GOTO 590

```

```

560 IF Z$="Y" P(5)=P(5)+1

```

```

570 PRINT "IS IT 'FROST FREE?'"
\ INPUT Z$
580 IF Z$="Y" P(5)=P(5)+2\ PRINT
"NEXT TIME BUY THE KIND YOU
DGFROST YOWRSENF AND SAVE ENER
GY"
680 PRINT "DOYOU HAVE EXCELLENT
INSULATION AND WEATHERSTRIPPING
, AND KEEP YOUR THERMOSTAT LOW?
"\ INPUT Z$

770 PRINT "DO YOU COMIONLY DRIV
E AU:      1)45MPH    2)55MPH
3)65MPH    4)70MPH    5)MOR?" \ I
NPUT Z
780 LET P(8)=P(8)*(1+(Z-2)/10)
785 IF Z>=3 CLEAR \ PRINT "YOU
KNOW IT TAKES MORE GASOLINE TO D
RIVE FAST;JR, DON'T YOU?"

830 PRINT "1 BACON,EGGS,TOAST,
COFFEE,      FRFNCH TOAST
2 PANCAKES,ORANGE J
UICE, MILK  3 MILK AND CEREAL
4 WATER AND DANDELI
ON GREFNS,  GRAP? FRUIT HALF OR O
RANGI?"

870 PRINT "NEXT, CONSIDER A DIN
NGR MGNU,   AND AGAIN MAKE YO'IR
CHOICE BY   TYPING THE MENU NUIB
,JR."
880 PRINT "1 STEAK,POTATOES,VE
GETABLE,CAKE COFFEE
2 GROUND BEFF PATTY
OR FISH,    CORN, SOUP, SALAD,PUDD
ING,MILK   3 BE IF STEW,SALAD,C
OFFEE     4 SALAD- FRUIT,NUTS,
TEA"\ INPUT Z

930 PRINT "TO PRODUCE MORE THAN
MINIMAL    FOOD YOU MUST INVEST
ENFRGY.    REMEIBIR THHAT I E.U
. IS THE   'NTIRE ENERGY USE OF
SOME PEOPLEFOR A WHOLE YEAR.HOW
MUCH ENRGYDO YOU WANT TO SP'ND
IN EACH    CATEGORY?"
931 PRINT "FFERTILIZATION,CULTIV
ATION,     RESEARCH AND D'VELOP
MGNT, AND  FOOD PROCESSING?"

1010 PRINT "INTO FOOD PROCESSIN
G(TO MAKE  POTATO CHIPS,CORN F
LAKES, FROZENFRIED,CANNED FOODS,
BREF....)" \ INPUT Z
1020 LET A(4)=Z
1030 LET Z=A(1)
1040 IF A(2)<A(1)Z=A(2)
1050 LET Z=0
1060 RETURN
1065 PRINT "IN ANSWFRING THE NE
XT QUESTIONS REMEMBER THAT A MAN
CAN LIVE A  WHOLE YEAR ON ONE E
U."

1370 IF E<0 PRINT "TO FILL ENER
GY DGFICIT IMPORJ";-B"UNITS OF
OIM." \J=E\0=0+E
1371 PRINT "SO2 IN AIR IS";B(2)

1380 IF E<0V=V+E/5000\E=0
1390 PRINT "THE DOLLAR IS WORTH
";V;\ POKE 240,0\ PRINT "YEAR";Y
\ POKE 240,2

1430 IF B(1)>99 PRINT \ PRINT \
PRINT "*** AIR IMPOSSIEL?***
*** YOU DIE! ***"

1480 IF B(2)>99 PRINT \ PRINT
\ PRINT "AIR IMPOSSIBLE! YOU DI
G"\ PRINT \ PRINT \ END
1490 IF B(2)>80 PRINT \ PRINT
"*THE AIR STINKS. I CAN HARD
LY BREATH?*" \ PRINT \ PRINT

1530 IF B(3)>99 PRINT \ PRINT
\ PRINT "****ALL DEAD FROM RA
DIATION****" \ PRINT \ END
1540 IF B(3)>80 PRINT \ PRINT "
RADIATION SF'RIO'IS. SOME DEAD!" \R
=R-B(3)/100\ PRINT \ PRINT

1625 PRINT "LAST TIME YOU USEF:
"\ POKE 240,0\ PRINT C(1);C(2);C
(3);\ POKE 240-2\ PRINT C(4);C(5
)
1630 PRINT "HOW MUCH INERGY WIL
L YOU TAKE FROM EACH SOURCE?(C
OAL,OIL....)" \ INPUT C(1),C(2),C
(3),C(4),C(5)

LAR IS WORTH";V;\ POKE 240,0\ PR
INT "YEAR:";Y\ POKE 240,2
1830 PRINT "CHOOSI:" \ INPUT Z
1835 IF Z>0 IF Z<9 GOTO 1870
1840 PRINT "0 ANOTHIR YEAR
1 RE-ENTER APPLIAN
CE ENJIRGY 2 RE-ENTER HEATING
3 RE-ENTER CAR
4 REGNTER FOOD"
1850 PRINT "5 AGRICULT'IRE
6 EDUC/FORESTPOPUL
ATION 7 DISPLAY SITUATIO
N 8 CHOICE OF ENERGY
SOURCES"

```







270

```

0390 GOSUB 640:GOSUB 650:GOSUB 650:GOTO 270
0400 GOSUB 640:GOSUB 690:GOSUB 690:GOTO 270
0410 GOSUB 640:A=C(CAS).75:GOSUB 700:A=C(CAS).25:GOSUB 650:A=C:G
OSUB 760
0420 LET A=C(CAS).25:GOSUB 710:A=C:GOTO 270
0430 GOSUB 640:GOSUB 710:GOSUB 640:GOTO 270
0440 GOSUB 640:GOTO 270
0450 GOSUB 750:GOSUB 740:GOSUB 640:GOTO 270
0460 GOSUB 640:GOSUB 860:GOTO 270
0470 GOSUB 640:GOSUB 740:GOSUB 740:GOTO 270
0480 GOSUB 485:GOSUB 640:A=C:GOSUB 890:GOSUB 880
0481 GOSUB 485:GOSUB 640:A=C:GOTO 270
0485 IF C(CGT)5 THEN LET A=A-2
0486 IF C(LT)5 THEN LET A=A-1
0487 RETURN
0490 GOSUB 640:GOSUB 890:GOSUB 640:GOTO 270
0500 GOSUB 640:GOSUB 700:GOSUB 640:GOTO 270
0510 GOSUB 640:GOSUB 690:GOSUB 790:GOTO 270
0520 GOSUB 640:A=C(CAS).75:GOSUB 700:A=C(CAS).25:GOSUB 650:A=C:G
OSUB 640
0530 LET A=C(CAS).25:GOSUB 710:A=C:GOTO 270
0540 GOSUB 640:GOSUB 690:A=C(CAS).5:GOSUB 640:GOSUB 810:A=C:GOTO
270
0550 GOSUB 770:GOSUB 650:GOSUB 760:GOTO 270
0560 GOSUB 780:GOSUB 640:GOSUB 780:GOTO 270
0570 GOSUB 640:GOSUB 740:GOSUB 640:GOTO 270
0580 GOSUB 890:GOSUB 880:GOTO 270
0585 GOSUB 790:GOSUB 690:GOSUB 640:GOTO 270
0590 GOSUB 485:GOSUB 640:A=C:GOSUB 880:GOSUB 890
0591 GOSUB 485:GOSUB 640:A=C:GOTO 270
0595 GOSUB 640:GOSUB 650:GOSUB 640:GOTO 270
0600 GOSUB 900:GOTO 270
0605 GOSUB 780:GOSUB 780:GOSUB 640:GOTO 270.
0610 GOSUB 970:GOSUB 800:GOSUB 960:GOTO 270
0615 GOSUB 640:GOSUB 650:GOSUB 760:GOTO 270
0620 GOSUB 920:GOTO 270
0625 GOSUB 700:GOSUB 650:A=C(CAS).5:GOSUB 640:A=C:GOSUB 810:GOTO
270
0630 GOSUB 940:GOTO 270
0635 GOSUB 790:GOSUB 710:GOSUB 640:GOTO 270
0640 FOR Y=1TOA:PRINT(CNO)K,TAB(CS);:FOR X=1TO10:PRINT(CNO)K,GE;:
NEXT X
0645 PRINT (CNO)K:NEXT Y:RETURN
0650 IF A(LT)1 THEN LET A=1
0670 FOR X=1TOA:PRINT(CNO)K,TAB(CS);GE;GE;HE;HE;GE;GE;HE;HE;GE;GE
:NEXT X
0671 RETURN
0680 PRINT (CNO)K,TAB(CS);
0690 FOR X=1TOA:PRINT(CNO)K,TAB(4(CAS)L+S);GE;GE;HE;HE;GE;GE:NEX
T X:RETURN
0700 FOR X=1TOA:PRINT(CNO)K,TAB(CS);GE;GE;:PRINT(CNO)K,TAB(8(CAS)

```

```

L+S); GE; GE;
0702 NEXT X
0704 RETURN
0710 IF A(LT)1 THEN LET A=1
0730 FOR X=1 TO A:PRINT(NOK,TAB(4(CAS)L+S));GE;GE;NEXT X:RETURN
0740 FOR X=1TOA:PRINT(NOK,TAB(S));GE;GE;NEXT X :RETURN
0750 FOR X=1TOA:PRINT(NOK,TAB(S));GE;GE;GE;GE;NEXT X:RETURN
0760 FOR Y=1TOA:PRINT(NOK,TAB(S));GE;GE;GE;GE;GE;GE;HE;HE;GE;GE
:NEXT Y
0761 RETURN
0770 FOR X=1TOA:PRINT(NOK,TAB(S));GE;GE;HE;HE;GE;GE;GE;GE;GE;GE
:NEXT X
0771 RETURN
0780 FOR X=1TOA:PRINT(NOK,TAB(8(CAS)L+S));GE;GE;NEXT X:RETURN
0790 FOR X=1TOA:PRINT(NOK,TAB(4(CAS)L+S));GE;GE;GE;GE;GE;GE;NEX
T X:RETURN
0800 FOR X=1TOA:PRINT(NOK,TAB(S));GE;GE;GE;GE;GE;GE;NEXT X:RETU
RN
0810 FOR X=1TOC/2:PRINT(NOK,TAB(S));:FOR Y=1TO2:FOR Z=1TO(5
(CAS)L)-X
0811 PRINT (NOK,GE(1));:NEXT Z
0820 IF X(GT)3 GOTO 950
0821 LET V=X
0830 PRINT (NOK,HE(V(CAS)2)
0831 NEXT Y
0832 IF L(LT)3 THEN PRINT(NOK)
0834 NEXT X:RETURN
0840 FOR X=1TOC/2:PRINT(NOK,TAB(S));HE(X));:FOR Y=1TO10(CAS)L-(2
(CAS)X)
0841 PRINT (NOK,GE(1));:NEXT Y:PRINT(NOK)
0850 NEXT X: RETURN
0860 FOR X=4(CAS)L TO 7(CAS)L STEP 14(CAS)L/(C(CAS)6)
0861 PRINT (NOK,TAB(X+S));GE;GE;GE;
0870 PRINT (NOK,TAB(7(CAS)L-X+S));GE;GE;GE;NEXT X:RETURN
0880 FOR X=0TO6(CAS)L STEP 6(CAS)L/C:PRINT(NOK,TAB(X+S));GE;GE;
GE;GE;NEXT X
0881 RETURN
0890 FOR X=6(CAS)L TO 0 STEP -6(CAS)L/C
0891 PRINT (NOK,TAB(X+S));GE;GE;GE;GE
0892 NEXT X
0893 RETURN
0900 FOR X=0TO7(CAS)L STEP(14(CAS)L)/(C(CAS)6):PRINT(NOK,TAB(X
+S));GE;GE;GE
0910 PRINT (NOK,TAB(7(CAS)L-X+S));GE;GE;GE;NEXT X :RETURN
0920 FOR X=0 TO 7(CAS)L STEP(7(CAS)L)/(C(CAS)3)
0921 PRINT (NOK,TAB(S));GE;GE;
0922 PRINT (NOK,TAB(X+S));GE;GE;GE;
0930 PRINT (NOK,TAB(8(CAS)L+S));GE;GE;NEXT X:RETURN
0940 FOR X=1 TO A(CAS)3:PRINT(NOK):NEXT X:RETURN
0950 FOR W=1TOX-3:PRINT(NOK,' ':NEXT W:V=3:GOTO 830
0960 FOR X=4(CAS)L TO 7(CAS)L STEP L(CAS)4/C
0961 PRINT (NOK,TAB(X+S));GE;GE;GE;NEXT X:RETURN
0970 FOR X=7(CAS)L TO 4(CAS)L STEP -L(CAS)4/C
0971 PRINT (NOK,TAB(X+S));GE;GE;GE;NEXT X:RETURN
0990 END

```

\*=(AS) #=£ ' >=(GT) <=(LT) #=(NO)

PATCH FOR PROGRAMMA SPHEREFORTH

As I reported previously, I have encountered certain annoying bugs in SphereFORTH. From reading the FORTH Interest Group Newsletter, it seems others are even less happy with this 6800 implementation. I find I have mislearned a few things that a "standard" FORTH would not have confused me with, like array handling. Nonetheless, I am still greatly enthusiastic concerning the FORTH approach to programming, and have found that fixing system bugs is not particularly hard, since these are just errors in defined words which are in the dictionary like any others.

I provide here the patches for the signed comparisons. (<, >) Adding replacement words to the dictionary would work for new (user) words, but would not retrofit the fix into earlier (system) words that rely on comparison, such as MIN and MAX. The crux of the problem was that the low-order byte of each value was being treated as if it were a signed value (like the more-significant byte, which properly is treated as signed). So, the unsigned comparisons BHI, BEQ, and BLS had to be substituted for the BGE and BLE used by Programma. Unfortunately, there is no BHS (Branch High or Same?), so that the usual patcher's nightmare of "just one extra instruction" was encountered, and a jump to the "common" code of the less-than routine was required.



T. R. Meier  
80 Alcott Street  
Acton, MA 01720

Entire routines are shown, but only 12 bytes are patched.

<u>Address</u>	<u>Original</u>	<u>New</u>
162D:	DE54 LDX stkptr	
	A601 LDAA 1,X	
	E600 LDAB 0,X	
	6F00 CLR 0,X	
	6F01 CLR 1,X	
	E002 SUBB 2,X	
	2E06 BGT .+6	
	2D06 BLT .+6	
	A003 SUBA 3,X	
163F:	<u>2F02</u> BLE .+2	2302 BLS .+2
	6C01 INC 1,X	
1643:	BD115F JSR 115F	
	BD100F JSR 100F	
	39 RTS	
...	...	
1650:	DE54 LDX stkptr	
	A601 LDAA 1,X	
	E600 LDAB 0,X	
	6F00 CLR 0,X	
	6F01 CLR 1,X	
	E002 SUBB 2,X	
165C:	<u>2D06</u> BLT .+6	2D08 BLT .+8
	<u>2E06</u> BGT .+6	2E08 BGT .+8
	A003 SUBA 3,X	
1662:	<u>2C02</u> BGE .+2	2204 BHI .+4
	<u>6C01</u> INC 1,X	2702 BEQ .+2
	<u>BD115F</u> JSR 115F	6C01 INC 1,X
	<u>BD100F</u> JSR 100F	7E1643 JMP 1643
	39 RTS	166B: xxxx spare

## COSMAC UT-4 Monitor Interface

This program lets the COSMAC UT-4 monitor think it is talking to a standard 300 band terminal instead of a Sphere System. All of the UT-4 commands are supported plus three additional functions.

- "I" will initialize the C.R.T. to a blank screen with the cursor in the upper left corner. It does not affect UT-4.
- "ESC" leaves UT-4 and returns control to the Sphere DEBUG mode.
- "T" requests a block of memory to be transferred between the COSMAC and Sphere System. The transfer program will ask for three numbers in the following format:
- "S=" start address in Sphere System,  
 "#=" number of bytes to be transferred, and  
 "C=" start address in COSMAC system.

After the three numbers have been entered, the program expects one of three commands.

- "L" will load the block of memory into the COSMAC from the Sphere System. Program control is returned to UT-4.
- "S" will store the block of memory into the Sphere System from the COSMAC. Program control is returned to DEBUG.
- "C" will return program control directly to UT-4 without transferring anything.

Any character except "L", "S", or "C" will initiate a new transfer sequence.

Gardner Road  
 West Kingston, R.I.  
 02892  
 March 29, 1979

I have designed a microprocessor control system using the RCA 1802 COS MAC. The 1802 was chosen because my application is oceanographic data logging and low power is a must.

The hardware is assembled on three different printed circuit boards. They are small enough to fit in a six inch I.D. cylindrical case. The CPU board has latches for the 8 high order address bits. It also has a power switched 2716 2K EPROM. There is a 4K CMOS RAM card which can be "piggybacked" to 8K. The third board has an 1854 UART and some miscellaneous I/O circuitry.

Since I did not have an RCA development system, I wrote the enclosed program to interface my Sphere System with RCA's UT-4 monitor.

Someone may find this useful in connecting a RCA Elf to a 6800.

Donald Dorson

COSMAC UT-4 Monitor Interface

Cassette file CN, 600-7A5, CK=1C. 9 January 1979  
2716 PROM 19 January 1979. Serial interface at F028.

```

START  C18E ED F032 Home cursor.
       C18E ED F038 Clear display.
       C18E E6 17 } Initialize ACIA: 7 bits, even parity & stop bit.
       C190 E7 F028 } RTS low.           ÷ 64 clock, 19.2 KHz = 300 baud.
       C191 E6 2A
       C195 E7 F028
NEXT    C198 DE 1C }
       C19A E3 23 } Cursor control
       C19C CE 12F2 }
       C19F 29
BLINK   C1A2 27 F5 }
       C1A2 7D F221 } Test keyboard
       C1A5 2A 1C }
       C1A7 E6 F222 Read character
       C1AA E1 19 "I" for initialize
       C1AC 27 EA START
       C1AE E1 1B "ESC" to return
       C1B2 E6 23 } to DEBUG
       C1B2 7E FE4F }
       C1B5 E1 54 "T" for transfer
       C1B7 27 48 TRANSFER
       C1B9 F6 F028 }
       C1BC 54 } TDRE?
       C1BD 54
       C1BE 24 F5 }
       C1C0 E7 F029 Transmit character
       C1C3 F6 F028 }
       C1C6 54 } RDRF?
       C1C7 24 DSBLINK
       C1C9 E6 F229 Read character
       C1CC 27 E1 BLINK Ignore Nulls
       C1CE E1 2A Test for L.F.
       C1D2 E6 26 } Teletype replacement
       C1D2 E6 16 } Initialize with line folding to
       C1D4 97 D0 } character counter accommodate 32 character
       C1D5 24 C7 BLINK } per line C.R.T.
       C1D8 7A 20D2 } Test for 25th character.
       C1DB 26 1F DISPLAY
       C1DD DE 1C } Make sure cursor is
       C1DF E6 52 } clear before C.R.
       C1E1 A7 00 }
       C1E3 5F }
       C1E4 E6 2D } Add 13 (decimal)
       C1E6 9E 1D } to cursor pointer.
       C1E8 97 1D }
       C1EA D9 1C }
       C1EC D7 1C }
       C1EE C1 E2 }
       C1F2 26 AD BLINK }
       C1F2 ED FD16 C.R.L.F. } If cursor goes off
       C1F3 CE E1E5 } the screen, reset to
       C1F6 DF 1C } bottom line.
       C1FA 22 A3 BLINK }
DISPLAY C1FC ED FCAD Display character.
NEXT 1 C1FF 22 97 NEXT (also used as island for return from "transfer".

```

D. L. Derson  
Narragansett Bay Campus URI  
South Ferry Road  
Narragansett, R. I. 02882  
U. S. A.

## COS MAC UT-4 Monitor Interface

## Memory Block Transfer Program

3

## Temporary memory usage

Start address 3C, 3D  
 # of bytes 3E, 3F  
 COSMAC address 3A, 3B

TRANSFER C201 BD FD16 C.R.L.F.

C204 86 53 "S"

C205 8D 1F A

C208 D7 3C

C20A 97 3D

C20C 8D FD16

"S = " Start address in Sphere System.

C20F 86 23 "#"

C211 8D 14 A

C213 D7 3E

C215 97 3F

C217 8D FD16

C21A 86 43 "C"

"# = " Number of bytes to be transferred.

C21C 8D 09 A

C21E D7 3A

C220 97 3B

C222 8D FD16

C225 20 2C B

A C227 8D FCAD

"C = " Start address in COSMAC system.

C22A 86 3D

C22C 8D FCAD

C22F 8D FEF4 INPNUM

C232 39

B C233 8D FE5C

C236 81 43 "C" for return to COSMAC UT-4

C238 27 C5 NEXT 1

C23A 81 4C "L" for load COSMAC from Sphere

C23C 27 06 LOAD

C23E 81 53 "S" for store in Sphere from COSMAC

C240 27 7C STORE

C242 20 8D TRANSFER

} Transfer command  
 sequence.

D. L. Dorson  
 Narragansett Bay Campus URI  
 South Ferry Road  
 Narragansett, R. I. 02882  
 U. S. A.





COSMAC UT-4 Monitor Interface

This program stores a block of memory into the Shpere System from the COSMAC. Program control returns to DEBUG.

```

STORE  C2EE 85 3F  "?"
        C2E8 8D CE  UT-4C
        C2C2 86 3E
        C2C4 8D 9E  HEXASC
        C2C6 86 3F
        C2C8 8D 57  HEXASC
        C2CA 86 2D  "C.R."
        C2CC 8D E4  TX
        C2CE 8D FD 15 C.R.L.F.
LINE   C2D1 8D E1  Rx
        C2E3 81 2A
        C2D5 25 FA
        C2D7 85 23
        C2D9 87 D2
        C2DB 8D FD 0F
        C2DE 8D 36
        C2E2 8D FC AD
        C2E3 8D CF
        C2E5 8D FC AD
        C2E8 7A 20 D2
        C2EB 25 F5
        C2ED 8F 1C
        C2EF 85 12
        C2F1 87 B1
DATA   C2F3 8E 3C
        C2F5 8D 1F  CHARACTER
        C2F7 48
        C2F8 48
        C2F9 48
        C2FA 48
        C2FB 35
        C2FC 8D 18  CHARACTER
        C2FE 33
        C2FF 1E
        C300 87 22
        C302 88
        C303 8F 3C
        C305 8E 3E
        C307 89
        C308 8F 3E
        C30A 25 03
        C30C 7E FE 4F  Return to DEBUG
        C30F 7A 02 D1
        C312 25 DF  DATA
        C314 22 8B  LINE
CHARACTER C316 8D 9C  Rx
        C318 81 46
        C31A 22 FA
        C31C 81 42
        C31E 22 0B
        C320 81 35
        C322 22 F2
        C324 81 2F
        C326 23 EE
        C328 82 32
        C32A 35
        C32B 82 37
        C32D 35
    
```

UT-4 query command.

# of bytes.

Close command with C.R.

Wait for line feed

Display address information.

Initialize data byte/line counter.

Pack 2 hex characters to make one byte.

Store one line of data. (16 bytes).

Store byte.

Test for end #.

Return to DEBUG

Test for last data byte in line.

Read characters from COSMAC and return with the first hex digit. The ASCII is converted to hex before return.

D. L. Dorson  
 Narragansett Bay Campus URI  
 South Ferry Road  
 Narragansett, R. I. 02882  
 U. S. A.

Bill J. Rutherford  
3201 N. Rd. 450 W.  
Kokomo, IN 46901

\*ADD A TELETYPE TO TINY BASIC<sup>†</sup>\*

YOU CAN INTERFACE A TELETYPE TO TINY BASIC USING THE ROUTINES ON THE FOLLOWING PAGES. ALSO, IF YOUR TELETYPE HAS A PAPER TAPE READER AND PUNCH, YOU CAN LOAD AND SAVE PROGRAMS THIS WAY. THE TTY SHOULD BE INTERFACED TO THE ACIA AT F050.

THERE ARE TWO FLAGS AT \$00FE AND \$00FF TO INDICATE WHICH PORTS ARE BEING USED FOR INPUT AND OUTPUT, RESPECTIVELY. A \$00 IN THE INPUT FLAG TELLS THE ROUTINE TO GET ITS INPUT FROM THE SPHERE KEYBOARD, AND AN \$FF TELLS IT TO ACCEPT INPUT FROM THE TTY KEYBOARD. LIKEWISE, A \$00 IN THE OUTPUT FLAG TELLS THE ROUTINE TO DISPLAY ITS OUTPUT ON THE CRT, AND AN \$FF DIRECTS THE OUTPUT TO THE TTY PRINTER. THUS, ONLY ONE TERMINAL CAN BE USED FOR INPUT OR OUTPUT AT A TIME. A CTRL/I CHANGES THE INPUT PORT USED, AND A CTRL/O CHANGES THE OUTPUT PORT.

THERE ARE TWO WAYS TO CHANGE THE I/O FLAGS FROM A BASIC PROGRAM. ONE, OF COURSE, IS TO CHANGE THEM WITH THE USR FUNCTION AND THE POKE SUBROUTINE. THE OTHER WAY IS TO USE THE USR FUNCTION TO CALL THE OUTPUT SUBROUTINE AT \$0209, PASSING THE ASCII CODE FOR A CTRL/I OR A CTRL/O TO THE SUBROUTINE IN THE THIRD ARGUMENT. THE STATEMENT

```
70 I=USR(521,0,9)
```

WILL CALL THE SUBROUTINE AT \$0209, WHICH JUMPS TO MY ROUTINE AT \$011D, WHERE THE CTRL/I PASSED IN THE THIRD ARGUMENT WILL BE PROCESSED (THE SECOND ARGUMENT IS MERELY A DUMMY). IN LIKE MANNER, THE STATEMENT

```
90 I=USR(521,0,15)
```

WILL "PRINT" A CTRL/O.

AFTER THE OCCURRENCE OF EITHER A CTRL/I OR CTRL/O IN THE INPUT STREAM, THE INPUT CHARACTER IN ACCUMULATOR A IS CHANGED TO A RUBOUT (HEX \$7F) WHICH IS IGNORED IN THE CHARACTER INPUT ROUTINE OF THE BASIC INTERPRETER. OTHERWISE, THE CONTROL CHARACTER WILL BE PLACED IN THE INPUT BUFFER, AND ANYTHING TYPED IN AFTER THAT WILL RESULT IN AN ERROR MESSAGE.

THE BREAK TEST ROUTINE CHECKS ONLY THE ACTIVE INPUT PORT FOR A BREAK. NULLS ARE NOT ACCEPTED AS BREAK CHARACTERS; EVERYTHING ELSE IS.

THE BACKSPACE CHARACTER HAS BEEN CHANGED TO THE ASCII UNDERLINE (SHIFT/O ON THE TELETYPE OR DELETE ON THE SPHERE KEYBOARD). THIS IS SO THE BACKSPACE CHARACTER WILL BE VISIBLE ON THE PRINTER. THE CURSOR POSITION IS STILL BACKED UP ON THE CRT, BUT NOW EACH ERRONEOUS CHARACTER IS CLEARED TO A SPACE.

TWO OTHER MODIFICATIONS HAVE BEEN MADE TO FACILITATE PAPER TAPE LOADING AND DUMPING. ONE IS THAT LOCATION \$0211, DEFINING THE NUMBER OF PAD CHARACTERS, HAS BEEN CHANGED TO A 2. THIS MEANS THAT AFTER A CARRIAGE RETURN, TWO NULLS WILL BE OUTPUT BEFORE TINY ECHOES WITH A LINEFEED AND ANOTHER NULL. ON PAPER TAPE, THE TWO PAD CHARACTERS INSURE THAT TINY WILL HAVE SUFFICIENT TIME TO PROCESS EACH LINE BEFORE THE TAPE ADVANCES TO THE NEXT LINE.

(INCIDENTALLY, ALTERING THE CONTENTS OF LOCATION \$0211 IS THE EQUIVALENT OF THE NULL FUNCTION IN SOME HIGHER LEVEL BASICS.)

ALSO, THE TAPE MODE HAS BEEN ENABLED BY CHANGING LOCATION \$0212 TO AN \$80. THIS WILL SUPPRESS ALL OUTPUT, PARTICULARLY LINE PROMPTS, WHILE A PAPER TAPE IS LOADING. THIS MODIFICATION IS ALSO MEANT TO GIVE TINY TIME TO PROCESS LINES. IF THE TAPE MODE IS DISABLED (LOCATION \$0212 CONTAINS A \$00), A PROMPT WILL BE PRINTED BEFORE EACH LINE, WHICH MAY CAUSE BASIC TO MISS THE FIRST FEW CHARACTERS OF THE NEXT LINE.

## TTY I/O DRIVER ROUTINES FOR TINY BASIC

00FE 00	INFLG FCB	0	0 SELECTS KBDPIA, \$FF SELECTS ACIA.
00FF 00	OUTFLG FCB	0	0 SELECTS CRT, \$FF SELECTS TTY.
020F 5F	BACKSP FCB	\$5F	
0211 02	PDCHRS FCB	2	
0212 80	TAPFLG FCB	\$80	
	*		
001C	CSRPTR EQU	\$1C	
0032	EDIT EQU	\$32	
00BF	TAPMOD EQU	\$BF	TAPE MODE IS ON WHEN SIGN BIT SET.
0300	CLDADR EQU	\$300	
0325	WRMADR EQU	\$325	
F040	KBDPIA EQU	\$F040	
F050	ACIA EQU	\$F050	
FC4A	GETCHR EQU	\$FC4A	
FCBC	PUTCHR EQU	\$FCBC	
	*		
	* I/O ROUTINES		
	*		
0100 8D 40	IN	BSR INCH	INPUT A CHARACTER INTO ACC. A
0102 B1 020F		CMPA BACKSP	
0105 26 0E		BNE OUT1	
0107 D6 FF		LDAB OUTFLG	OUTPUT TO CRT?
0109 2B 4A		BMI OUTTTY	NO, PRINT BACKSPACE CHAR. ON TTY
010B DE 1C		LDX CSRPTR	
010D 09		DEX	BACK UP CURSOR ON THE SCREEN
010E DF 1C		STX CSRPTR	
0110 C6 20		LDAB =\$20	
0112 E7 00		STAB 0,X	BLANK OUT ERRONEOUS CHARACTER
0114 39	EXIT1	RTS	
0115 81 0D	OUT1	CMPA =\$0D	
0117 26 04		BNE OUT	
0119 D6 BF		LDAB TAPMOD	DON'T PRINT CR IF TAPE MODE OFF
011B 2A F7		BPL EXIT1	
011D 81 09	OUT	CMPA =9	
011F 26 05		BNE OUT2	
0121 73 00FE		COM INFLG	CHANGE THE INPUT PORT
0124 20 07		BRA CLRCHR	
0126 81 0F	OUT2	CMPA =\$0F	
0128 26 06		BNE OUT3	
012A 73 00FF		COM OUTFLG	CHANGE THE OUTPUT MEDIUM
012D 86 7F	CLRCHR	LDAA =\$7F	KEEP CTRL CHAR OUT OF INPUT BUFFER
012F 39	EXIT2	RTS	
0130 D6 FF	OUT3	LDAB OUIFLG	OUTPUT TO CRT?
0132 2B 21		BMI OUTTTY	NO, PRINT CHAR. ON TTY
0134 81 0A		CMPA =\$0A	DON'T PRINT CURSOR & CTRL CHARS ON
0136 2F F7		BLE EXIT2	THE CRT.



## DIGITAL TAPE SYSTEM

## REQUIREMENTS:

1. Low cost
2. High reliability
3. Fast (4800 BAUD or more)
4. Easy interface to Sphere
5. Software compatible to Sphere (Any changes transparent to standard Sphere cassette and my CC8A manual digital Cas.)

## THE MICRO COMMUNICATIONS CORP. MODEL 25-3000 TAPE SYSTEM:

The Microcom is a very small tape transport with attached PC board electronics. The PC board provides full TTL compatible circuitry for control and reading and writing on endless loop Wafer-cartridges. The wafers use cr02 on a mylar base and are available in 5 to 50 feet lengths. The system sells for \$110. and delivery (in April) was instant. The transport without electronics sells for \$40. , and the special motor regulator IC is also available. Very good documentation includes complete schematics, timing charts and flowcharts.

The wafers are 100% certified and cost about \$24. per case of 10 dependent on length. Write protect conductive 'stick-ons' are provided.

The Microcom with some extra circuitry meets all 'requirements' NICELY.

## THE ADDED CIRCUITRY:

1. Permits automatic fast search (10 "/S) while reading at 3"/S.
2. Hardware write protect
3. Overwrite error stop (requiring manual reset) when writing and end of tape occurs. Since the tape wafer is an endless loop, this protects from overwriting accidentally with a 'too short' tape.
4. Beginning of tape positioner switch for fast parking.
5. Power supply circuitry that eliminates the possibility of writing a flux change during power loss or system shutdown.

(Note; a 6.8K resistor is paralalled with R22 of the microm PC board for 10"/S fast speed)

The system performs errorless at 4800 B and with a 25 feet tape you can be reading any block in the 30 kbyte capacity in 30 seconds or less.

NOTE: 9600 B operation in my system yields an error value of approximately 1 bit in  $10^6$  while at 4800 B no errors occur; therefore satisfactory operation should extent to 7000 or 8000 B.

This is faster operation than some 9600 Baud systems due to the 3.3:1 search/read ratio and approximately 30 mS fast to slow transition time. This permits speeding through nearly all of the interrecord gap before slowing down to test the block name.

I have been using the system for about 2 months in breadboard form and have found it to be an excellent performer. The system will operate without fast search with no software changes, however with about 66 bytes additional eeprom the fast search provisions can be used and is well worth the effort.

## SIM board Mods. (See Re articles)

The SIM board should be crystal controlled for maximum reliability. Other very minor changes and fixes are required.

The oscillator of Vol 1 ISSUE 3 may be used with appropriate circuit values:

XTAL	153.6 kHz (to 9600 Baud)
R1,R3	100k
R2	47k
R4,R5	47k
C1,C2	not used

(Output is collector of Q3)

Complete schematics for the CAS1 version are available on request. An unused CAS2 could probably be modified without much trouble.

## SOFTWARE:

Computable software driver changes and additions are enclosed showing the added code starting at FA00, however RDHRY and RDHDR may be changed to suite your eprom space.

I think the Microcom is a very good system and well worth the minimum cost and effort required. Although Micro Comm. Corp. is essentially an OEM supplier, I had no trouble purchasing 4 units (through Airep Electronics Inc., Dallas, TX)

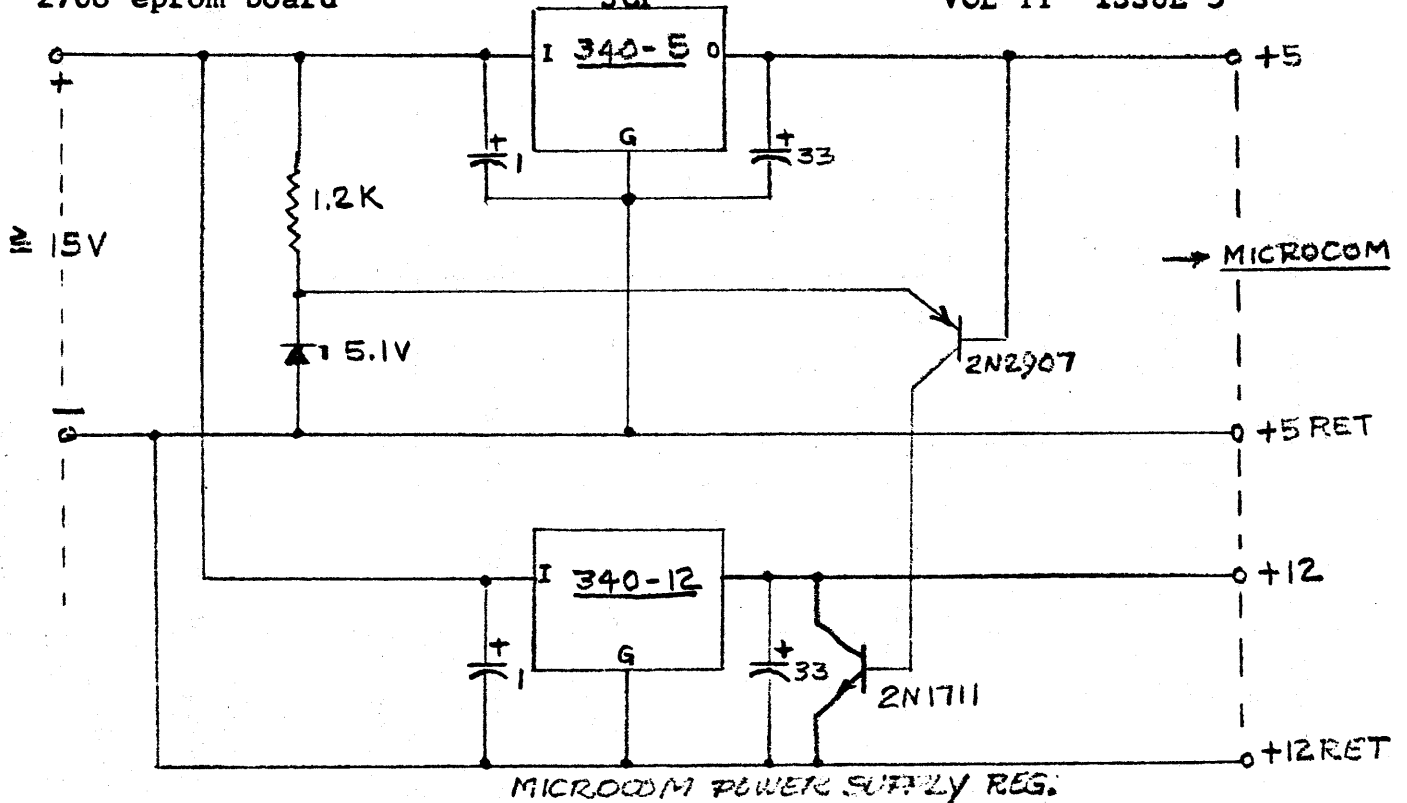
The information presented here is very sparse; feel free to write for any additional information you may require.

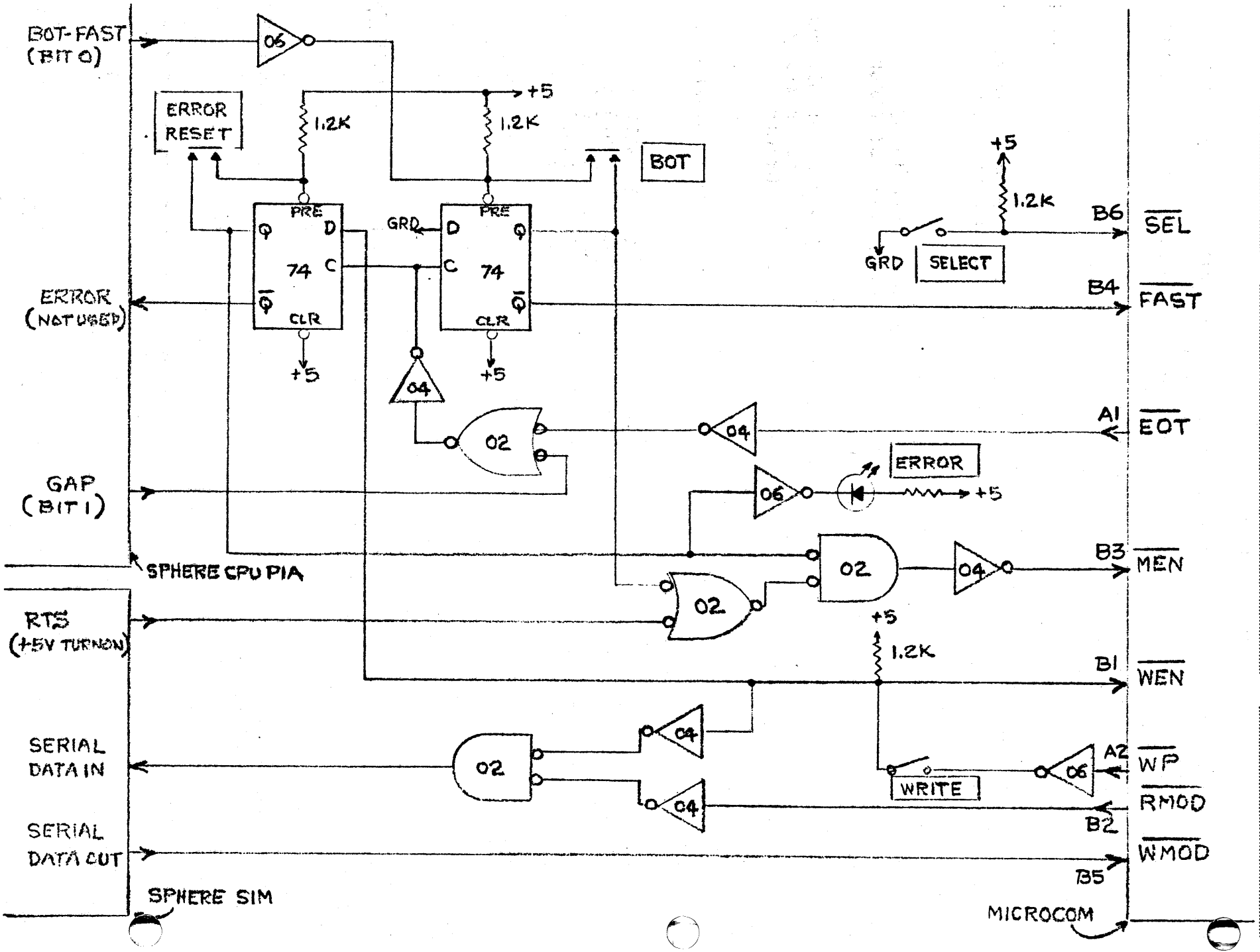
## REF. INFORMATION:

Interfacing CC-8a  
Crystal control SIM  
Oscillator Divider  
Add 1702A to SIM  
2708 eprom board

Jim Raehl  
Jim Raehl  
JCP  
Jim Raehl  
JCP

VOL 111 ISSUE 1  
VOL 111 ISSUE 1  
VOL 1 ISSUE 3  
VOL 1 ISSUE 1  
VOL 11 ISSUE 3







## PAGE NO 01 MICROCOM CASSETTE DRIVER

```

0001          *
0002          *
0003          * IR GAP LOCATER
0004          *
0005 0200          ORG    SFBB5
0006          *
0007          *RDH DRY IS THE LOCATION OF
0008          *ADDED EPROM
0009          *TIME IS HEX NUMBER OF LOOPS
0010          * AT APPROX 50 MICROSEC/LOOP
0011          *
0012  FBB5          TIME    EQU    $3000    (LOOPS, 50US)
0013  FBB5          RDH DRY EQU    $FA00    ADD EPROM LOC
0014  FBB5          RDH DR EQU    $FA03    EPROM +3
0015          *
0016  FBB5          ACIA NO EQU    $38
0017  FBB5          COUNT  EQU    $18
0018  FBB5          OUTREG EQU    $F042    PIA OUTPUT R
0019  FBB5          CONREG EQU    $F043    PIA CONTROL R
0020          *
0021          *
0022  FEB5 7E FA00 RDH DRX JMP    RDH DRY
0023          *
0024  FEB8 86 01    IRGAP  LDAA  =01        FAST
0025  FEB8 B7 F042          STAA  OUTREG
0026  FEBD CE 3000  IRGAP1 LDX   =TIME
0027  FEC0 DF 18          STX   COUNT
0028  FEC2 DE 38          LDX   ACIA NO
0029  FEC4 6D 01          TST   1,X      CLR FULL BIT
0030  FEC6 DE 18          INPUT LDX   COUNT
0031  FEC8 27 0D          BEG   GAP
0032  FECA 09          DEX
0033  FECB DF 18          STX   COUNT
0034  FECD DE 38          LDX   ACIA NO
0035  FECF 86 01          LDAA =501    TST FOR CHR
0036  FED1 A5 00          BITA 0,X
0037  FED3 27 F1          NOGAP BEG   INPUT    NO CHR YET
0038  FEB5 20 E6          BRA   IRGAP1   NOT TIME OUT
0039  FED7 86 02          GAP   LDAA  =02
0040  FED9 B7 F042          STAA  OUTREG    SLOW
0041  FEDC 7F FC42          CLR   OUTREG    LEAVE LOW
0042  FEDF 7E FA03          END   JMP    RDH DR

```

```

0001      *
0002      *
0003      *LEAD HEADER MOD AND INITIALIZE
0004      *FOR MICROCOM CASSETTE DRIVER
0005      *
0006      C200          ORG      $FA00
0007      *ORIGEN IS LOC. OF ADDED EPROM
0008      *
0009      FA00          IRGAP    EQU      $FBEB
0010      FA00          CASIN    EQU      $FB7E
0011      FA00          BLKNAM   EQU      $33
0012      FA00          BLKN+1  EQU      $34
0013      FA00          NOPRNT   EQU      $3A
0014      FA00          BFRPTR   EQU      $3C
0015      FA00          BFRP+1   EQU      $3D
0016      FA00          BFRSZE   EQU      $3E
0017      FA00          BFRS+1   EQU      $3F
0018      FA00          SYNC     EQU      $16
0019      FA00          ESC      EQU      $1B
0020      *
0021      *
0022      FA00 7E FBEB  RDHDRY   JMP      IRGAP
0023      FA03 8E FB7E  RDHDR   JSR      CASIN
0024      FA06 81 16    RDHDR1  CMPA    = SYNC
0025      FA08 26 F9    BNE     RDHDR
0026      FA0A BD FB7E  JSR     CASIN
0027      FA0D 81 1E    CMPA    = ESC
0028      FA0F 26 F5    BNE     RDHDR1
0029      FA11 8E FB7E  JSR     CASIN
0030      FA14 16      TAB
0031      FA15 BD FB7E  JSR     CASIN
0032      FA18 9E 3D    ADDA    BFRP+1
0033      FA1A D9 3C    ADCE    BFRPTR
0034      FA1C 97 3F    STAA   BFRS+1
0035      FA1E D7 3E    STAB   BFRSZE
0036      FA20 8E FB7E  JSR     CASIN
0037      FA23 36      PSHA
0038      FA24 BD FB7E  JSR     CASIN
0039      FA27 D6 3A    LDAB   NOPRNT
0040      FA29 33      PULB
0041      FA2A 27 06    BEQ    RDHDR2
0042      FA2C F7 E03E  STAB   $E03E
0043      FA2F B7 E03F  STAA   $E03F
0044      FA32 7D C033  RDHDR2 TST    BLKNAM
0045      FA35 27 0B    BEQ    RDHDR3
0046      FA37 D1 33    CMPE   BLKNAM
0047      FA39 26 04    BNE    BADNAM
0048      FA3E 91 34    CMPA   BLKN+1
0049      FA3D 27 03    ELQ    RDHDR3
0050      FA3F 7E FBEB  BADNAM JMP    IRGAP
0051      FA42 39      RDHDR3 RTS
0052      *
0053      *
0054      *
0055      *
0056      *
0057      *
0058      *

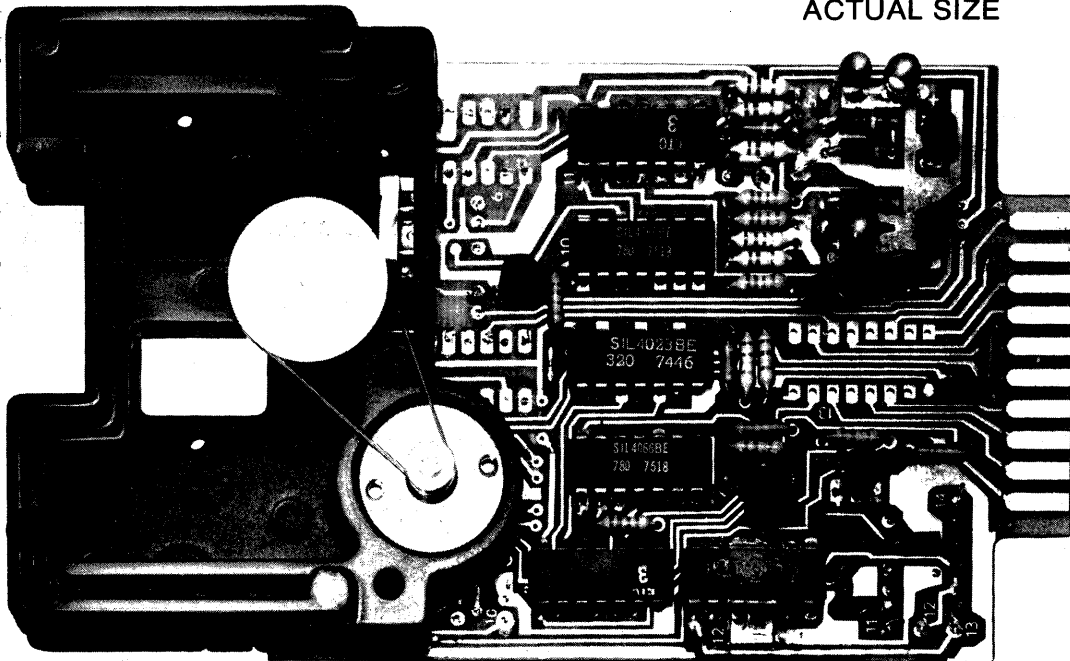
```

```

0059          *
0060          *
0061          *INITIALIZE
0062          *
0063 FA43      OFF      EQU      $51
0064 FA43      ACIANO   EQU      $38
0065 FA43      OUTREG   EQU      $F042
0066 FA43      CONREG   EQU      $F043
0067          *
0068 FA43 CE 0000 INTLZX   LDX      =0000
0069 FA46 FF F042      STX      OUTREG
0070 FA49 86 FF        LDAA     =$FF
0071 FA4E B7 F042      STAA     OUTREG
0072 FA4E 86 04        LDAA     =$04
0073 FA50 B7 F043      STAA     CONREG
0074          *
0075 FA53 DE 38        INTLZ1  LDX      ACIANO
0076 FA55 86 13        LDAA     =$13
0077 FA57 39          ENDX     RTS
0078          *
0079          *
0080          *REPLACE CODE AT FB00 WITH
0081          *THE FOLLOWING:
0082 FA58          ORG      $FB00
0083          *
0084 FE00 BD FA43      INTLZ   JSR      INTLZX
0085 FE03 A7 00        STAA     0,X
0086 FE05 86 51        LDAA     =OFF
0087 FE07 A7 0C        STAA     0,X
0088 FE09 39          RTS

```

ACTUAL SIZE



## Electronic Read/Write Tape System

The Micro Read/Write Digital Data Storage System is a complete digital input/output system designed for use in storing and retrieving digital information that is encoded and decoded by the user. In addition to complete electronics, the Electronic Read/Write System contains sensors for both EOT/BOT and Write Permit. When coupled with a micro-processor or other CPU (or the appropriate hardware logic) an Electronic Read/Write System becomes a complete serial data storage and retrieval system at a fraction of the usual cost. Because they leave the encoding and decoding to the user, Electronic Read/Write Systems permit the use of any bit-serial, self-clocking code.

COMPUTERWARE  
1512 Encinitas Blvd  
ENCINITAS, CALIFORNIA 92024

NOTE: IF ENOUGH OF  
US ARE INTERESTED  
WE CAN BULK  
PURCHASE BASIC FROM  
CSS. CONTACT JEFF

July 27, 1979

Dr. Jeffrey Brownstein  
2 Tor Road  
Wappingers Falls, NY 12590

Dear Dr. Brownstein,

We have been receiving customer orders for cassette BASIC from sphere owners who write that they have been referred by you, who will provide the patches they need.

First, we would like to thank you for referring the orders to us and providing the service to the sphere owners. We appreciate your honesty and support!

Secondly, we would like to offer you a "dealer" type price schedule on copies of cassette BASIC if you would like to purchase them for resale directly.

<u>Quantity</u>	<u>Discount</u>	<u>Copy Price</u>
2-3	25%	\$22.50
4-5	30%	\$21.00
6-10	35%	\$19.50

Of course, we will continue to fill any orders we receive and will inform sphere owners to your service.

Thank you for your continued interest and support.

Sincerely,

*Sue Searby*  
Sue Searby  
Computerware

SCOTT ADAMS GRAPHICS SOFTWARE.....

We Have complete listings for the software for Scott Adams Graphics Boards. They will not publish well, please contact Jeff Brownstein for individual copies.

Roger and other SPHERE users,

After reading the SPHERE CRT-80 controller, I realized that it had the makings of a powerful multiple/split screen editor driver controller. When I inspected the code closely, I even found that the modifications generally speeded it up and didn't cost any extra memory (trading some in-line bytes for some in page 0).

By making CRTBEG, CRTEND, LINSIZ, LNCTR, and SCRLFG all page 0 locations instead of equates or program constants, a 9byte SCREEN CONTROL BLOCK is created (which also includes CSRPTR) which can be filled and changed by your program to give, at your command different screen areas within the screen memory area as a whole.

This also allows the same code to be used with different sized screens, just by changing the parameters in the control block. Now the code can go into a PROM and not have to be changed when you upgrade your 16 x 32.

The changes to the code published are pretty simple:

1. Remove all '#'s for the above variables
2. Create page 0 labels for all the above variables,
3. Recode the portion at CRT36A, where LINSIZ is used as an index offset. The following should work

```

CRT36A   JSR      PSHX
         LDA B    LINSIZ
         BSR      CRT033
         CPX      CRTEND
         BEQ      CRT36B
         LDA A    0, X
         JSR      PULX
         STA A    0, X
         INX
         BRA      CRT36A
CRT36B   JSR      PULX
         ( BRA      CRT034 )

```

There are various places that the code can be improved by minor changes; JSR's followed by RTS's, some JSR's become BSR's by simple reordering, and if CRT036 thru CRT36B were put in front of CRT034 it would be able to fall through.

Good luck, and thanks for all the work that goes into the newsletter.

*John Rible*  
 John Rible  
 10 Fairfield St.  
 Cambridge, MA 02140

*Was a very poor typist - sorry*

# **ATTENTION:**

## **RENEWALS**

If you have not renewed your subscription to SPHERE, this issue was sent to you FREE. Please do not make this one your last one.

**RENEW NOW:**

\$12 Domestic and Canada

\$16 Foreign

**SEND TO:**

Jeff Brownstein  
2 Tor Road  
Wappingers, NY. 12590

## **NEXT ISSUE**

FINALLY, the 64 Character Mod to the SPHERE video screen. That's right, 64 characters. The mod will cost about \$15 dollars, it has already been done, it works, and it takes about 1-2 hours to do.

FORTH: 12 pages of SPHERE FORTH information. Mods, Patches, New Words, Expanded dictionary listing.