SPHERE

VOL IV ISSUE 4

MICROCOMPUTER USERS' NEWSLETTER
FEBRUARY 1980

EDITORS: JEFF BROWNSTEIN
ROGER J. SPOTT

SOFTWARE

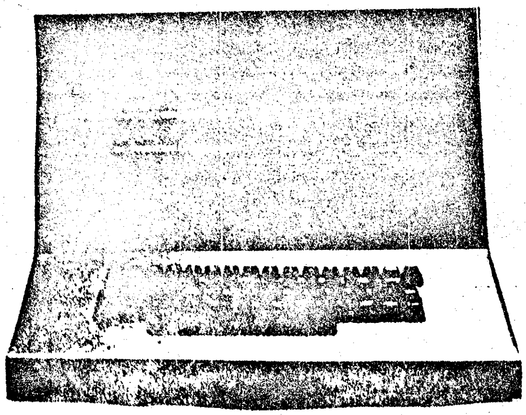
*****WE WILL NEED MORE MATERIAL IN ORDER TO PRODUCE THE *****

NEXT NEWSLETTER

HARDWARE

Please Send Typed Material For Next Issue TO:

Jeff Brownstein
2 Tor Road
Wappingers, N.Y.
12590
or
Roger Spott
13975 Connecticut Ave.
Wheaton, Maryland 20906



SPHERE PASCAL

Both Roger and I are working on patching a pretty well documented version of Pascal for cassette users. From the letter that follows, there is at least one other Sphere user working on this. Please let us know who you are so we can join forces.

I have also included in the newsletter a few sheets of the instruction manual so that you can get the feel of the capabilities of this particular implementation. A complete "supervisor" program and "editor" are part of the package. The supervisor handles operating commands like LOAD, SAVE, EDIT, COMPILE,GO, MOVE and QUIT. The editor uses fairly standard means to help in creating the Pascal source. The commands are: NEW, TOP,BOTTOM,UP,DOWN,FIND,PRINT,INSERT,KILL,REPLACE and QUIT.

Roger and I have ideas on how to handle floating point with the package although we have some work to do before addressing that need. Please let us know if you are at all interested in Pascal.

Jeff

EDITOR'S NOTE

We have tried to start even the late subscription renewals with the August issue as this keeps our bookkeeping simpler. If you renewed late and failed to receive the August, October or the December issue, please let us know.

Editors

Dyna
soft
systems

P.O. BOX 51
WINDSOR JCT., N.S.
CANADA B0N 2V0
(902) 861-2202

January 14, 1980

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

Dear Dr. Brownstein:

Thank you for your order for a copy of DYNASOFT PASCAL. A program cassette and user's manual are enclosed.

Appendix B of the manual should contain enough information to allow you to patch my package to run on the Sphere computer, but I have also enclosed the source code for the p-code interpreter, which is the only part of the system that interfaces with external input-output routines. The program was assembled with a general-purpose machine-independent cross-assembler, and the mnemonics are somewhat different from Motorola's, but the translation should be obvious. The rest of the system is in pseudo-code and contains only one reference to the "outside world": a jump (in p-code) to the main entry point for MIKBUG/SWTBUG which implements the SUPERVISOR "Quit" command (see the bottom of page B-2). I would prefer not to release the source code for the compiler itself at this time.

I would be very pleased to have you publish your patches in the Sphere Journal, provided that I am correct in my interpretation that it is only the patches (not the entire package) that you intend to publish. It would probably bring me another wave of new orders, for which I would of course be grateful. Incidentally, the very first order I received was also from a Sphere user.

If you have any difficulties adapting the package, I can be reached most evenings at the above number. In any case, please let me know how you make out. I would be particularly interested in seeing any patches for a disk environment, since I have not yet had a chance to address that problem myself.

Please note that in spite of what Byte magazine says, I am located in Nova Scotia, not North Saskatchewan.

Yours truly,

Allan G. Jost

Allan G. Jost, Ph.D.

rc
Enclosures

I. THE DYNASOFT PASCAL LANGUAGE

The language PASCAL was originally conceived by Professor Niklaus Wirth at the Institut fuer Informatik, ETH Zurich, Switzerland, in 1968. It was designed to be a vehicle for the teaching of programming as a systematic discipline, but it has proven to be highly suited to a variety of applications and it has gained a wide following. While there have been several successful implementations of the full language on microcomputer systems, they typically require the order of 48K of memory and a disk drive, and this has limited their use to relatively large and expensive systems. One of the reasons for the success of BASIC on microcomputers has been the wide availability of practical implementations which will run on small scale cassette-based systems. DYNASOFT PASCAL was designed to make a workable implementation of PASCAL available to this class of small scale systems using the 6800 microprocessor. It is based on a subset of standard PASCAL which includes most of the standard control structures but omits some of the more sophisticated data structuring features and floating point arithmetic. The result is a complete high-level language program development system which retains most of the flavour and structure of standard PASCAL, but which will run on a system with as little as 12K bytes of memory and a single cassette drive.

The complete definition of standard PASCAL is contained in the book "PASCAL: User Manual and Report", by Kathleen Jensen and Niklaus Wirth, which is recommended reading for anyone who is seriously interested in the PASCAL language. What follows here is a summary of the DYNASOFT subset of standard PASCAL.

Vocabulary

The basic symbols of DYNASOFT PASCAL consist of the upper case letters, the digits 0-9, certain special characters and character combinations, and a set of keyword or reserved words which have special meaning to the compiler. The reserved words are:

AND ARRAY BEGIN CASE CONST DIV
DO DOWNTO ELSE END EXTERN FOR
FORWARD FUNCTION IF MOD NOT OF
OR OTHERWISE PROCEDURE PROGRAM READ REPEAT
THEN TO TYPE UNTIL VAR WHILE
WRITE WRITELN

These words may not be used for any purpose other than that intended in the definition of the language, which means in particular that they may not be used as identifiers.

Identifiers are user-defined names which denote constants, variables, types, procedures or functions, and consist of a letter followed by any combination of letters and digits, the first four of which must be unique. Since the compiler uses only the first four characters of words, no identifier may have its first four characters the same as any reserved word.

There is a set of identifiers which are pre-declared in DYNASOFT PASCAL. These are all names of pre-defined data types or standard procedures or functions. These names are not reserved words, and they may be re-declared by the user if desired. The pre-declared identifiers are:

INTEGER CHAR ODD SHL SHR HALT
LINK MOVL MOVR FIND SETP

The special symbols are:

+ - := . , *

; : = <> <= <

> >= ( ) [ ]

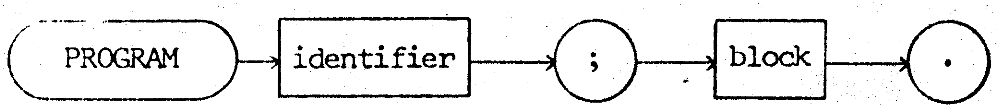
(* *) ..

Comments begin with the character pair '(*' and end with the first subsequent appearance of the character pair '*)'. Comments may appear anywhere a blank may appear, except embedded in a literal character string. The compiler listing option is controlled by a special form of comment. A comment beginning with the sequence '(*$L-' suppresses source code listing (except for error lines), and the sequence '(*$L+' turns the listing option back on.

Character strings are enclosed in single quotes, and may be up to 80 characters long. It is not possible to imbed quote marks in a string.

Syntax:

The syntax rules of PASCAL will be illustrated with a graphical aid known as the syntax diagram, which will require some explanation. A diagram contains two kinds of symbols: terminal symbols and non-terminal symbols. Terminal symbols are written in round or oval enclosures, and consist of keywords and special characters which have special meaning in the language. Non-terminal symbols are written in rectangular boxes, and are the names of other diagrams containing a further expansion of the syntax. The valid paths through a syntax diagram are indicated by directional arrows. The diagram for the non-terminal symbol program is shown below. Starting with program, and repeatedly expanding the non-terminals using other diagrams, any path through the syntax diagrams represents a syntactically valid program.



(cont)

MATRIX OPERATORS IN CSS BASIC

It is always possible to write matrix operations using Basic by using for-next loops to load, operate on, and print out an array. One of the reasons that this is sometimes less than ideal is that it is really difficult to see what the triple and quadruple nested loops are doing when you look at a program. Also, the programs take a long time to write and debug. They run slower than programs using matrix operators directly.

Probably this project (which is applicable to all of the Uiterwik style 6800 Basics) will be entered in the 6800 JOURNAL program contest if I finish it in time. Meanwhile, all but two modules of the group are running. The package consists of:

MATPRINT Prints out a specified array to screen or other port
MATTAB Tabs the printed array away from the left of screen
MATSPREAD Spreads out the printed array for easier viewing

MATREAD Reads in an array from Data statements
MATSET Sets the whole array equal to some number. Replaces MAT ZER and MAT CON which are limited to setting to only zero or one.

MATIDN Sets up an identity matrix according to your dimensions
MATSCALARLET ADD, subtract, multiply or divide each element by a Scalar (variable).
MATSCALARLET A(1,1)= B(1,1) + X
MATLET Add, subtract, multiply and divide each element by the same element of a second array and place the results in a third array.
MATLET A(1,1)= B(1,1) / C(1,1)
Note: This is not true algebraic matrix multiplication or division. There is no true algebraic matrix division and algebraic multiplication is handled in this package by MATPRODUCT

MATTRANS Changes the array around. An example: A(2,9) is changed into the new array B(9,2). The old array remains intact

MATPRODUCT Does algebraic matrix multiplication (THIS MODULE NOT FINISHED)

MATINVERSE Calculates the inverse matrix (THIS MODULE NOT FINISHED)

Note that a matrix is called by its first member. This will usually be as follows: A(0,0) or A(1,1) depending on whether Base= has been used.
Also, it is possible to use MATREAD, MATPRINT, and MATTAB to handle a one dimension list like B(80)

```
         DE LDX 30     Save number build buffer pointer
         DF STX 6C     In 6C
         DE LDX 2C     Get Basic source pointer
         BD JSR 12B0   Call read routine
         DF STX 2C     Save new Basic Source pointer
         DE LDX 63     Basic variable pointer loaded in X
         09 DEX
         09 DEX
         EE LDX 00     Backup in variable table
         DF STX 6E     Store in 6E where the variable ends
DATA     96 LDAA 63    Move the info into number build buffer
         D6 LDAB 64
         DE LDX 6C
         A7 STAAX00
         E7 STABX01
         DE LDX 44     Get Data Pointer
         9C CPX 20     Compare to beginning of program
         27 BEQ 18     TOKN
LOOP     BD JSR 0F0D   Put data into number build buffer
         24 BCC 1D     READ
         A6 LDAAX00
         27 BEQ 05     LOOP
ERR      C6 LDAB#19    error 19
         7E JMP 0FE7
LOOP1    9C CPX 22     Compare to end of program
         27 BEQ F7     ERR
         A6 LDAAX00
         08 INX
         4D TSTA       Test for end of line
         26 BNE F6     LOOP1
TOKN     A6 LDAAX03
         81 CMPA#C8    Look for "DATA" in token form
         26 BNE F0     LOOP1
         8D BSR 1F     INX5
         20 BRA DE     LOOP
READ     BD JSR 032B   Put some data into variable table
         A6 LDAAX00
         81 CMPA#2C    look for comma
         26 BNE 01
         08 INX        Bypass comma
STOR     DF STX 44     Store new data pointer
         BD JSR 04AC   Reset number build buffer pointer
         DE LDX 63
         8D BSR 09     INX6   Advance to next storage in variable table
         9C CPX 6E     End of array?
         26 BNE 01
         39 RTS        Exit from MATREAD
CONT     DF STX 63
         20 BRA B2     DATA
INX6     08 INX        Subroutine to increment the index register
INX5     08 INX        thus advancing in the variable table
         08 INX
         08 INX
         08 INX
         08 INX
         39 RTS
```

*****MATREAD is relocatable. All of the modules are complete individual additions. You could add to your CSS Basic any or all of them. Use of common subroutines could shorten the package though. My latest cassette version is complete with the matrix operators except MATPRODUCT and and MATINVERSE which are not completed.

```
             * * * * * * * * * * * * * * * *
             *
             * DISASSEMBLED LISTER ROUTINE
             *
             * PRODUCES FORMATTED HEX DUMPS
             *
             *
5000                       ORG  $5000
             *
5000                  ACIA EQU  $F050   PRINTER PORT
             *
5000                  KBDD EQU  $F040   KEYBOARD DATA
5000                  DBUG EQU  $FE64   PDSV3NF DBUGER
5000                  CURS EQU  $1C     CRT CURSOR
5000                  EDTF EQU  $37     EDITOR FLAG
5000                  ON   EQU  $A      SETS UP ACIA
5000                  OFF  EQU  $4A     SETS DSR
5000                  RSET EQU  $13     RESETS ACIA
5000                  ACK  EQU  6       ANSWERBACK
             *
             *
             *
5000 BD5103   STRT JSR  INIT    INITZ I/O PORT
5003 CE513E        LDX  #FF$    FORM FEED
5006 BD50D2        JSR  PDA1
     *
5009 CE514E   LIST LDX  #ORG$   START ADDR MSG
500C BD50D2        JSR  PDA1
500F BD509F        JSR  BADR    GET START ADDR
5012 CE5148   NEXT LDX  #EOL$   NEW LINE
5015 BD50D2        JSR  PDA1
5018 CE515D   NXT1 LDX  #XHI    X:=XHI
501B BD50FB        JSR  04HS    PRINT ADDR
501E FE515D        LDX  XHI     GET ADR POINTR
5021 A600          LDAA 0,X     GRAB BYTE
5023 B75159        STAA TEMP
5026 BD50FD        JSR  02HS    PRINT DATA
5029 FF515D        STX  XHI
502C BD50FF        JSR  OUTS    PRINT SPACE
502F 5F            CLRB
5030 B65159        LDAA TEMP    RESTORE A
5033 818C          CMPA #$8C    OP CODE 'CPX'?
5035 2718          BEQ  OP.3    THREE BYTES
5037 818E          CMPA #$8E    OP CODE 'LDS'?
5039 2714          BEQ  OP.3
503B 81CE          CMPA #$CE    OP CODE 'LDX'?
503D 2710          BEQ  OP.3
503F 84F0          ANDA #$F0    MASK OFF
5041 8120          CMPA #$20    BRANCH?
5043 270B          BEQ  OP.2
5045 8160          CMPA #$60    LESS THAN $60?
5047 2508          BCS  OP.1
5049 8430          ANDA #$30    MASK BITS 4&5
504B 8130          CMPA #$30    BOTH SET?
504D 2601          BNE  OP.2
504F 5C       OP.3 INCB
5050 5C       OP.2 INCB
5051 F7515A   OP.1 STAB BOX     PUT # IN BOX
5054 2708          BEQ  GO
5056 C101          CMPB #1      SET SPACECOUNT
5058 2702          BEQ  ONE1
505A CB01          ADDB #1
505C CB02     ONE1 ADDB #2
505E F7515C   GO   STAB SPCC    SPACECOUNT
5061 F6515A        LDAB BOX     RESTORE B
5064 2710          BEQ  NXT1    NEXT INSTRUCTN
5066 7A515A        DEC  BOX
5069 2705          BEQ  NLY1    PRINT NXT BYTE
```

```
506B BD50FB        JSR  04HS    PRINT 2 OF 3
506E 2003          BRA  SADR    SAVEADDR.
5070 BD50FD NLY1   JSR  02HS    PRINTBYTE & SP
5073 FF515D SADR   STX  XHI     POINT TO NEXT
5076 BC5162 NXTI   CPX  ENDP    DONE ?
5079 2603          BNE  NTST
507B 7EFE64 BUGJ   JMP  DBUG    GOTO DEBUGGER
507E 7DF040 NTST   TST  KBDD    REPEAT?.
5081 2BF8          BMI  BUGJ
5083 B6515B        LDAA LINC    GET LINE COUNT
5086 8403          ANDA #3      MASK DOWN
5088 2605          BNE  NXTC    UPDATE COLUMN#
508A 7C515B        INC  LINC    BUMP LINECOUNT
508D 2083          BRA  NEXT    NEW LINE.
508F 7C515B NXTC   INC  LINC    BUMP COUNTER
5092 C607          LDAB #7
5094 F0515C        SUBB SPCC    UPDATE SP CNT
5097 8D66   SPCL   BSR  OUTS    OUTPUT SPACE
5099 5A            DECB
509A 26FB          BNE  SPCL    SPACE LOOP
509C 7E5018        JMP  NXT1    NEXT-SAME LINE
            *
            * I/O SUBROUTINES
            *
            * BUILD ADDRESS
509F 8D0C   BADR   BSR  BYTE    READ 2 FRAMES
50A1 B7515D        STAA XHI
50A4 8D07          BSR  BYTE
50A6 B7515E        STAA XLOW
50A9 FE515D        LDX  XHI     X GETS ADDR
50AC 39            RTS
            * INPUT BYTE
50AD 8D2E   BYTE   BSR  IHEX    GET HEX CHAR
50AF 48            ASLA         MOVE OVER
50B0 48            ASLA
50B1 48            ASLA
50B2 48            ASLA
50B3 16            TAB          SAVE IN B
50B4 8D27          BSR  IHEX    GET SECOND CH
50B6 1B            ABA
50B7 16            TAB
50B8 FB515F        ADDB CKSM
50BB F7515F        STAB CKSM
50BE 39            RTS
            * OUT HEX LEFT BCD DIGIT
50BF 44     OTHL   LSRA         MOVE OVER
50C0 44            LSRA
50C1 44            LSRA
50C2 44            LSRA
            * OUT RIGHT BCD DIGIT
50C3 840F   OTHR   ANDA #$F
50C5 8B30          ADDA #$30    MAKE ASCII
50C7 8139          CMPA #$39
50C9 234D          BLS  OUTC
50CB 8B07          ADDA #7
50CD 2049          BRA  OUTC
            * PRINT DATA POINTED AT BY X
50CF 8D47   PDA2   BSR  OUTC
50D1 08            INX
50D2 A600   PDA1   LDAA 0,X
50D4 8104          CMPA #4      EOT?
50D6 26F7          BNE  PDA2
50D8 39            RTS
            *
50D9 863F   GOOF   LDAA #'?
50DB 8D3B          BSR  OUTC
            * INPUT HEX CHARACTER
50DD 8D4D   IHEX   BSR  INCH
50DF 8030          SUBA #$30
```

```
50E1 2BF6          BMI  GOOF
50E3 8109          CMPA #9
50E5 2F0A          BLE  IN1H
50E7 8111          CMPA #$11
50E9 2BEE          BMI  GOOF
50EB 8116          CMPA #$16
50ED 2EEA          BGT  GOOF
50EF 8007          SUBA #7
50F1 39       IN1H RTS
 ..      * OUTPUT 2 HEX CHAR
50F2 A600     OT2H LDAA 0,X
50F4 8DC9     O2HA BSR  OTHL
50F6 A600          LDAA 0,X
50F8 08            INX
50F9 20C8          BRA  OTHR
 ..      * OUTPUT 4 HEX CHAR & SPACE
50FB 8DF5     O4HS BSR  OT2H
 ..      * OUTPUT 2 HEX CHAR & SPACE
50FD 8DF3     O2HS BSR  OT2H
 ..      * OUTPUT A SPACE
50FF 8620     OUTS LDAA #32   ASCII SPACE.
5101 2015          BRA  OUTC
 ..      *
 ..      * TURNS ON TERMINAL AND WAITS
 ..      * FOR MOTORS TO GET UP TO SPEED.
 ..      *
5103 CEF050 INIT LDX  #ACIA   GET PORT ADDR
5106 8613        LDAA #RSET   RESET ACIA
5108 A700        STAA 0,X
510A 860A        LDAA #ON     BRING UP DSR
510C A700        STAA 0,X
510E 8603        LDAA #3      TIME COUNT
5110 09     TLUP DEX
5111 26FD        BNE  TLUP
5113 4A          DECA
5114 26FA        BNE  TLUP
5116 8606   SACK LDAA #ACK    SELECT TERMINL
 ..      *
 ..      *
 ..      * SENDS CHAR IN A AT 150 BAUD
 ..      *
5118 FF5160 OUTC STX  XTMP    SAVE REGISTERS
511B 36          PSHA
511C CEF050      LDX  #ACIA   ASYNC PORT#2
511F 8602   OLUP LDAA #2      TEST MASK
5121 A400        ANDA 0,X     ACIA BUF. MPTY?
5123 27FA        BEQ  OLUP    LOOP UNTIL OK
5125 32          PULA         GET ORIG CHAR
5126 A701        STAA 1,X     PUT IN OUT BUF
5128 FE5160 OEXT LDX  XTMP    RESTORE INDEX
512B 39          RTS          AND RETURN
 ..      *
 ..      * INPUTS CHARACTER INTO A
 ..      *
512C FF5160 INCH STX  XTMP    SAVE INDEX
512F CEF050      LDX  #ACIA   GET ACIA ADDR
5132 8601   ILUP LDAA #1      TEST BIT MASK
5134 A400        ANDA 0,X     ANYTHNG THERE?
5136 27FA        BEQ  ILUP    TRY AGAIN
5138 A601        LDAA 1,X     LOAD IN CHAR
513A 847F        ANDA #$7F    STRIP PARITY
513C 20EA        BRA  OEXT    RESTORE & EXIT
 ..      *
         *
513E 0A0A   FF$  FDB  $A0A
5140 0A0A        FDB  $A0A    FORM FEED
5142 0A0A        FDB  $A0A
5144 0A0A        FDB  $A0A
5146 0A0A        FDB  $A0A
5148 0D0D   EOL$ FDB  $D0D
```

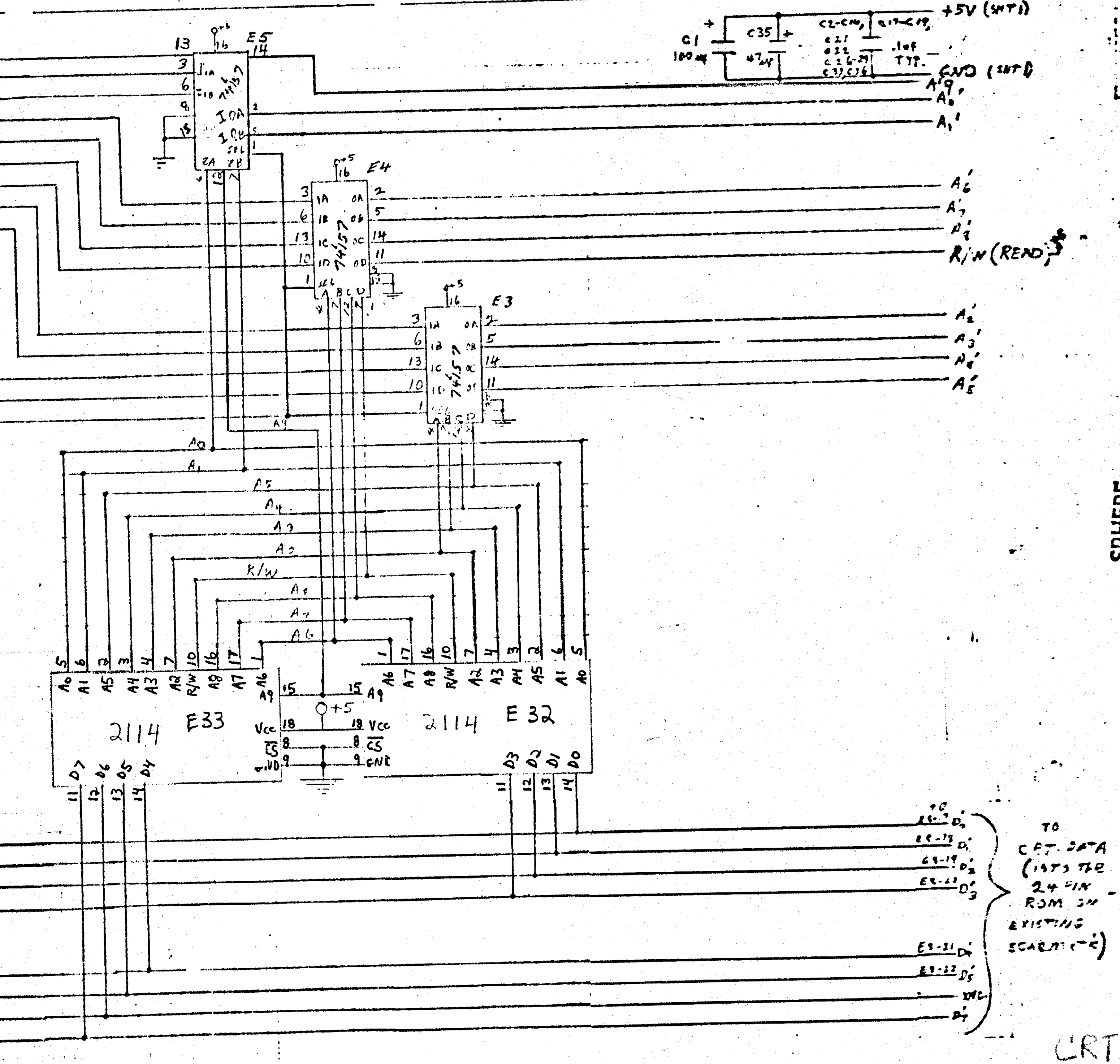
```
207 514A 0A00         FDB  $A00
208 514C 0004         FDB  4
209        *
210 514E 0D0A    ORG$ FDB  $D0A
211 5150 0053         FDB  $53
212 5152 5441         FDB  $5441
213 5154 5254         FDB  $5254
214 5156 2024         FDB  $2024
215 5158 04           FCB  4
216        *
217        *
218 5159         TEMP RMB  1      ACCA TEMP
219 515A         BOX  RMB  1      OP TYPE BOX
220 515B         LINC RMB  1      LINE COUNT
221 515C         SPCC RMB  1      SPACES COUNT
222 515D         XHI  RMB  1      ADDRESS.
223 515E         XLOW RMB  1
224 515F         CKSM RMB  1
225 5160         XTMP RMB  2
226 5162         ENDP RMB  2
227        *
228        *
229 5164         ENDS END
                 ERR=0


  SYMBOL TABLE:

ACIA F050  KBDD F040  DBUG FE64
CURS 001C  EDTF 0037  ON   000A
OFF  004A  RSET 0013  ACK  0006
STRT 5000  LIST 5009  NEXT 5012
NXT1 5018  OP.3 504F  OP.2 5050
OP.1 5051  ONE1 505C  GO   505E
NLY1 5070  SADR 5073  NXTI 5076
BUGJ 507B  NTST 507E  NXTC 508F
SPCL 5097  BADR 509F  BYTE 50AD
OTHL 50BF  OTHR 50C3  PDA2 50CF
PDA1 50D2  GOOF 50D9  IHEX 50DD
IN1H 50F1  OT2H 50F2  O2HA 50F4
O4HS 50FB  O2HS 50FD  OUTS 50FF
INIT 5103  TLUP 5110  SACK 5116
OUTC 5118  OLUP 511F  OEXT 5128
INCH 512C  ILUP 5132  FF$  513E
EOL$ 5148  ORG$ 514E  TEMP 5159
BOX  515A  LINC 515B  SPCC 515C
XHI  515D  XLOW 515E  CKSM 515F
XTMP 5160  ENDP 5162  ENDS 5164
```

SUBJECT CRT/1 Module SHEET NO. 2 OF 3

JOB NO. 1



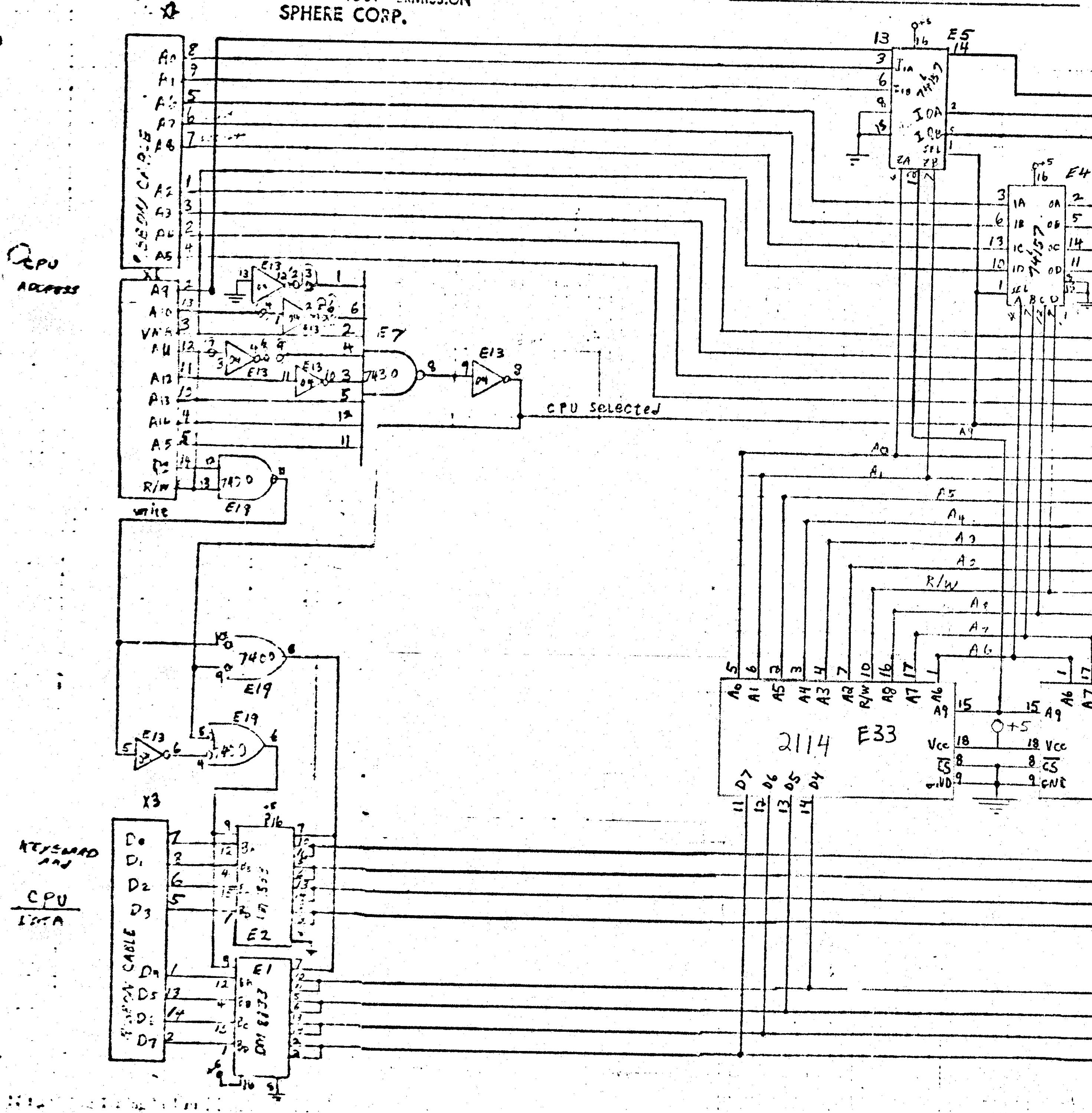
CRT/1

SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.

BY mct/SGE DATE 15 APR 75
CHKD. BY mct DATE 7/1/75

SUBJECT CRT/1 Module



BIORHYTHM CALENDAR WITH NAME AND DATE FILES (FOR CSS BASIC)

```
0001 REM *** NAME & DATE FILE SET-UP (ND) V-1 USING CSS SUPER BASIC V-4
0005 LET X=0
0006 INPUT "NUMBER OF INPUTS",Z
0010 INPUT "INPUT FILE NAME",F$
0020 OPEN O F$
0025 INPUT "NAME",N$
0030 INPUT "DATE MM,DD,YYYY",M,D,Y
0035 TWRITE N$
0040 TWRITE M,D,Y
0050 CLOSE
0060 LET X=X+1
0070 IF X>=Z THEN END
0080 GOTO 20
0090 END
0100 LET X=0
0105 INPUT "NUMBER OF OUTPUTS",Z
0110 INPUT "INPUT FILE NAME",F$
0120 OPEN I F$
0125 TREAD N$
0130 TREAD M,D,Y
0135 PRINT N$,M;D;Y
0140 PRINT #0,N$,M;D;Y
0150 LET X=X+1
0155 PRINT X
0160 IF X>=Z THEN END
0170 GOTO 120
0180 END
```

TO SET-UP FILE DO A RUN
TO PRINT LIST OF NAMES IN FILE DO A GOTO 100

SAMPLE PRINT-OUT
OF NAME/DATE FILE

```
G.K.HALE          6 13 1932
M.LONG            1 20 1920
J.MARTIN          5 7 1956
J.STANFORD        7 19 1947
S.ROBERTSON       12 17 1963
J.PETERS          1 17 1950
S.BULLINS         2 15 1951
C.PAUL            7 27 1934
A.PETERS          1 2 1952
S.FRANKLIN        1 4 1951
H.MILLER          10 18 1947
S.KATES           5 1 1945
G.RUDISILL        8 27 1937
C.ROBERTSON       9 17 1946
E.STEVENS         3 4 1932
```

The Biorhythm program can call the name and birth-date from the tape file or can be run 1 name and date at a time input from the keyboard.

SAMPLE PRINT-OUT
OF BIORHYTHM

COMPUTER GENERATED
BIORHYTHM CALENDAR FOR G.K.HALE

```
OCT                                                   OCT
1979      DOWN             CRITICAL              UP   1979
         -....................0....................+
1    M                      I E              P        1   M
2    TU               I     .   E             P     2   TU
3    W               I        .        E          P   3   W
4    TH           I           .           E       P   4   TH
5    F         I              .               E  P    5   F
6    SA      I                .               P E     6   SA
7    SU    I                  .            P     E    7   SU
8    M    I                   .       P           E   8   M
9    TU   I                   .  P               E    9   TU
10   W     I               P  .                 E     10  W
11   TH     I         P       .               E       11  TH
12   F        I  P            .           E           12  F
13   SA       P I             .        E              13  SA
14   SU    P       I          .   E                   14  SU
15   M    P            I      E                       15  M
16   TU   P               I   .                       16  TU
17   W      P        E        I                       17  W
18   TH        P  E           .   I                   18  TH
19   F        E     P         .      I                19  F
20   SA     E            P    .          I            20  SA
21   SU    E                  P             I         21  SU
22   M    E                   .    P          I       22  M
23   TU    E                  .         P       I     23  TU
24   W      E                 .              P   I    24  W
25   TH       E               .                 P I   25  TH
26   F            E           .                   I   26  F
27   SA             E         .                  IP   27  SA
28   SU                  E    .                I P    28  SU
29   M                        E              IP       29  M
30   TU                       .   E       IP          30  TU
31   W                        .       PI              31  W
         -....................0....................+
```

23 DAY PHYSICAL CYCLE = P 28 DAY EMOTIONAL CYCLE = E 33 DAY INTELLECTUAL CYCLE = I

THIS IS ONLY A DEMONSTRATION OF WHAT A COMPUTER CAN DO.
THIS COMPUTER IS USUALLY DOING ENGINEERING CALCULATIONS.

COMPLIMENTS OF ---
LONG ENGINEERING CO. - WINSTON-SALEM, N.C.

```
0010 REM *** BIORHYTHM (BD) V-2 USING CSS SUPER BASIC V-4
0020 LINE= 0
0100 DIM P1(23),P2(23),E1(28),E2(28),I1(33),I2(33)
0110 DIM W$(7),L$(41),M$(12)
0120 DATA SU,M,TU,W,TH,F,SA
0130 DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
0135 PRINT "         BIORHYTHM CALENDAR"
0136 PRINT
0137 PRINT "COMPUTING"
0140 FOR N=1 TO 23
0150 LET P2(N)=SIN(6.283185307*(N-1)/23)
0160 LET P1(N)=INT(21.5+20*P2(N))
0170 NEXT N
0180 FOR N=1 TO 28
0190 LET E2(N)=SIN(6.283185307*(N-1)/28)
0200 LET E1(N)=INT(21.5+20*E2(N))
0210 NEXT N
0220 FOR N=1 TO 33
0230 LET I2(N)=SIN(6.283185307*(N-1)/33)
0240 LET I1(N)=INT(21.5+20*I2(N))
0250 NEXT N
0260 FOR N=1 TO 7
0270 READ W$(N)
0280 NEXT N
0290 FOR N=1 TO 12
0300 READ M$(N)
0310 NEXT N
0320 FOR N=1 TO 41
0330 LET L$(N)=" "
0340 NEXT N
0345 INPUT "TYPE OF INPUT (SINGLE OR FILE)",Z$
0346 IF Z$="FILE" THEN GOTO 350
0347 IF Z$="SINGLE" THEN GOTO 400
0350 INPUT "NUMBER OF BIORHYTHMS",Z
0355 LET F$="BIO DATES"
0360 PRINT "ENTER MONTH & YEAR FOR BIORHYTHM (MM,YYYY)"
0365 INPUT M4,Y4
0366 LET X=0
0370 OPEN I F$
0375 TREAD N$
0380 TREAD M,D,Y
0390 GOTO 430
0400 PRINT "ENTER NAME"
0401 INPUT N$
0402 PRINT "ENTER BIRTH DATE (MM,DD,YYYY)"
0403 INPUT M,D,Y
0407 PRINT "ENTER MONTH & YEAR FOR BIORHYTHM (MM,YYYY)"
0420 INPUT M4,Y4
0430 GOSUB 1350
0440 LET M=M4
0450 LET D=1
0460 LET Y=Y4
0470 GOSUB 980
0480 LET S1=J
0490 GOSUB 1350
0500 LET L1=31
0510 IF M4=12 THEN 570
0520 GOSUB 1110
0530 LET S3=N3
0540 LET M=M4+1
0550 GOSUB 1110
0560 LET L1=N3-S3
```

```
0570 LET B=J-S1+1
0571 LET E=B+L1-1
0572 PRINT #0,CHR$(01)
0573 PRINT #0,TAB(5);"COMPUTER GENERATED"
0581 PRINT #0,TAB(5);"BIORHYTHM CALENDAR FOR ";N$
0582 SKIP #0,2
0585 PRINT #0,CHR$(02)
0590 PRINT #0,TAB(10);M$(M4);TAB(64);M$(M4)
0596 PRINT #0,TAB(10);
0600 PRINT #0,Y,TAB(20);"DOWN";TAB(37);"CRITICAL";TAB(59);"UP";TAB(64);Y
0610 PRINT #0,TAB(19);"-....................0";
0620 PRINT #0,"....................+"
0630 LET V=0
0640 FOR I=B TO E
0650 LET V=V+1
0660 LET J3=I-1
0670 LET K1=INT(J3/23)
0680 LET K2=J3-(K1*23)+1
0690 LET K3=INT(J3/28)
0700 LET K4=J3-(K3*28)+1
0710 LET K5=INT(J3/33)
0720 LET K6=J3-(K5*33)+1
0730 LET P=P1(K2)
0740 LET Q=E1(K4)
0750 LET R=I1(K6)
0780 LET L$(21)="."
0790 LET L$(P)="P"
0800 LET L$(Q)="E"
0810 LET L$(R)="I"
0820 PRINT #0,TAB(10);V;TAB(15);W$(N2);TAB(20);
0830 FOR N=1 TO 41
0840 PRINT #0,L$(N);
0850 NEXT N
0860 PRINT #0,TAB(64);V;TAB(68);W$(N2)
0870 LET L$(P)=" "
0880 LET L$(Q)=" "
0890 LET L$(R)=" "
0900 LET N2=N2+1
0910 IF N2<8 THEN 930
0920 LET N2=1
0930 NEXT I
0931 PRINT #0,TAB(19);"-....................0";
0932 PRINT #0,"....................+"
0934 PRINT #0,CHR$(30)
0935 PRINT #0,TAB(6);"23 DAY PHYSICAL CYCLE = P  28 DAY EMOTIONAL ";
0936 PRINT #0,"CYCLE = E  33 DAY INTELLECTUAL CYCLE = I"
0938 PRINT #0,CHR$(29)
0939 SKIP #0,1
0940 IF V=28 THEN GOSUB 1451
0941 IF V=29 THEN GOSUB 1455
0942 IF V=30 THEN GOSUB 1458
0946 SKIP #0,2
0948 GOSUB 1460
0950 SKIP #0,12
0955 IF Z$="SINGLE" THEN GOTO 970
0959 LET X=X+1
0960 PRINT X;"OF ";Z;"PRINTED"
0961 IF X>=Z THEN GOTO 963
0962 GOTO 370
0963 INPUT "RUN AGAIN",X$
0964 RESTORE
0965 IF X$="YES" THEN GOTO 350
```

```
0966 IF X$="STOP" THEN GOTO 1490
0967 IF X$="SINGLE" THEN GOTO 969
0968 GOTO 963
0969 LET Z$=X$: GOTO 400
0970 INPUT "RUN AGAIN",X$
0971 RESTORE
0972 IF X$="YES" THEN GOTO 400
0973 IF X$="SAME" THEN GOTO 407
0974 IF X$="STOP" THEN GOTO 1490
0975 IF X$="FILE" THEN GOTO 977
0976 GOTO 970
0977 LET Z$=X$: GOTO 350
0980 IF M<3 THEN 1020
0990 LET M1=M-2
1000 LET Y1=Y
1010 GOTO 1040
1020 LET M1=M+10
1030 LET Y1=Y-1
1040 LET C=INT(Y1/100)
1050 LET D1=Y1-(C*100)
1060 LET N4=INT((13*M1-1)/5)+D+D1+INT(D1/4)
1070 LET N=N4+INT(C/4)-2*C+77
1080 LET N1=INT(N/7)
1090 LET N2=N-(N1*7)+1
1100 RETURN
1110 LET Y2=INT(Y/4)
1120 LET Y3=Y-(Y2*4)
1130 IF Y3=0 THEN 1150
1140 GOTO 1250
1150 LET Y2=INT(Y/100)
1160 LET Y3=Y-(Y2*100)
1170 IF Y3=0 THEN 1190
1180 GOTO 1230
1190 LET Y2=INT(Y/400)
1200 LET Y3=Y-(Y2*400)
1210 IF Y3=0 THEN 1230
1220 GOTO 1250
1230 LET L1=1
1240 GOTO 1260
1250 LET L1=0
1260 LET N1=INT((3055*(M+2))/100)-91
1270 LET L=0
1280 IF M<3 THEN 1330
1290 IF L1=0 THEN 1320
1300 LET L=1
1310 GOTO 1330
1320 LET L=2
1330 LET N3=N1+D-L
1340 RETURN
1350 IF M<3 THEN 1390
1360 LET M1=M-3
1370 LET Y1=Y
1380 GOTO 1410
1390 LET M1=M+9
1400 LET Y1=Y-1
1410 LET C=INT(Y1/100)
1420 LET D1=Y1-(C*100)
1430 LET N=INT((146907*C)/4)+D+INT((1461*D1)/4)
1440 LET J=N+1721119+INT((153*M1+2)/5)
1450 RETURN
1451 SKIP #0,3
1454 RETURN
1455 SKIP #0,2
```

cont'd PAGE 24

LIBRARY

We have been fortunate enough to gain access to the Amateur Computer Group of New Jersey 6800 Software Library. A listing of the available material follows. I have transposed all of this stuff into Sphere cassette format so a certain amount of time is required to dump the programs for you. Please do not ask for it all at once. The basic programs are in our CSS format but also run on SWTP 8K Basic interpreters. If your interpreter reads the programs in Ascii let me know; our CSS version 4 reads both Ascii (LMIK#5) and Sphere formats. The Sphere binary code loads much faster so it is preferred. One must own a copy of 101 Computer Games or Some Common Basic Programs in order to have documentation for running the programs from them.

J. B.

(INTRODUC.TXT - DISK #6 VOL. 11)

THE PROGRAMS ON THIS SIDE OF THIS DISK ARE MUSIC PROGRAMS. THE MUSIC PROGRAMS ARE EQUIVALENT TO THE 6502 VERSION OF HAL CHAMBERLAIN'S MUSIC PROGRAMS IN THE SEPTEMBER 1977 ISSUE OF BYTE MAGAZINE.

THESE PROGRAMS ALLOW YOU TO PLAY 4 PART HARMONY ON THE 6800. THE MUSIC BOARD WHICH IS USED FOR THIS IS ONE WHICH UTILIZES TWO MC1408L8 ICS. THE CIRCUITRY ON THE MUSIC BOARD INCLUDES A SET OF FLIP FLOPS TO LATCH THE DATA, AND THE CIRCUIT FOUND IN THE SPEC SHEET APPLICATIONS ON THE MC1408. THE SPEC SHEET IS IN THE MOTOROLA LIBRARY BOOKS ON LINEAR DEVICES, AND THE CIRCUIT USED IS SHOWN IN FIGURE 35 ON PAGE 5-66. THIS CAN BE LEAD OUT THROUGH THE ACTIVE FILTER DESCRIBED IN HAL CHAMBERLAIN'S ARTICLE TO AN AUDIO AMPLIFIER CHIP.

PRESENTLY, THE CIRUIT IS ON A PROTOTYPE CARD, BUT A BOARD WILL BE LAYED OUT FOR THIS CIRCUIT AND INFORMATION WILL BE FURNINSHED LATER.

IT IS POSSIBLE TO USE THE MUSIC PROGRAMS WITH THE NEWTECH MUSIC BOARD, BUT PLEASE NOTE THAT THE NEWTECH BOARD USES ONLY 6 BITS FOR THE MUSIC WORD, WHILE THE CHAMBERLAIN USES 8 BITS. THE DIFFERENCE IS THAT THE NEWTECH DOES NOT DECODE THE VOLUME AND INTENSITY BITS, SO YOU WILL HAVE TO RUN YOUR NEWTECH BOARD THROUGH AN AMPLIFER, SINCE THE SOUND OUTPUT WILL BE LOW. EVEN THOUGH IT SOUNDS LOW IN VOLUME, IT IS A HELL OF A LOT BETTER THAN THE SOUND OF THE NEWTECH PROGRAMS.

TWO TEXTS ON THIS DISK CONTAIN THE SOURCE FOR THE MUSIC PROGRAMS. ONE IS CALLED 'MUSIC.TXT' AND THE OTHER IS 'PLAY.TXT' . THESE PROGRAMS ARE IN THE TSC ASSEMBLER FORMAT IF YOU WOULD LIKE A SOURCE LISTING.

THERE ARE TWO SONGS ON THIS DISK AS DEMOS: THE ENTERTAINER AND THE STAR SPANGLED BANNER. OTHER MUSIC PIECES INCLUDE SOME SOUND EFFECTS AND OARTS OF MUSIC AS SAMPLES.

Disk #1 - "101 Games"

```
           CATALOG OF DRIVE #1    Vol 1
            NAME   TYPE  SIZE

NJ-1   005 - CONTENTS.TXT     5                 TEXT EDITOR

       019 - ACEYDUCY.BAS    17                 C55BASIC (all except otherwise noted)
       058 - AMAZING .BAS    25
       109 - ANIMAL  .BAS    16
       140 - AWARI   .BAS    19
       175 - BAGELS  .BAS    17
       204 - BASEBALL.BAS    69
       295 - BANNER  .BAS    25
       328 - BASKTBAL.BAS    46
       385 - BATNUM  .BAS    18
       408 - BATTLE  .BAS    42
       457 - BINGO   .BAS    32
       493 - BLACKJK1.BAS    60
       535 - BLACKJK2.BAS    20
       576 - BOAT    .BAS    31
       605 - BOMBARD .BAS    22
NJ-1R  005 - BOMBAWY .BAS    23
       049 - BOUNCE  .BAS    12
       075 - BOWLING .BAS    26
       124 - BOXING  .BAS    27
       171 - BULLSEYE.BAS    14
       194 - BUNNY   .BAS    10

             SECTORS LEFT = 0


           CATALOG OF DRIVE #1    Vol 2
            NAME   TYPE  SIZE

       212 - BUG     .BAS    47
       281 - BULLFGHT.BAS    41
       335 - BUZWORD .BAS    11
       350 - CHANGE  .BAS    11
       366 - CHECKERS.BAS    28
       400 - CALENDAR.BAS    11
       415 - CHEMIST .BAS     9
       428 - CHIEF   .BAS    13
       444 - CHOMP   .BAS    18
       465 - CIVLWAR .BAS    96
       559 - COMBAT  .BAS    26
       586 - CRAPS   .BAS    17
NJ-2   005 - CUBE    .BAS    33
       072 - DEPCHARG.BAS    15
       103 - DIGITS  .BAS    20
       140 - DIAMOND .BAS     5
       152 - DICE    .BAS     6
       165 - EVNWIN1P.BAS    23
       204 - EVNWINS2.BAS    17
       232 - FTBALL1 .BAS    60
       315 - FOOTBALL.BAS    55

             SECTORS LEFT = 9
```

Disk #2 - "101 Games"

```
         CATALOG OF DRIVE #1    Vol 3
          NAME    TYPE  SIZE

NJ-2 382-FLIPFLOP.BAS    16
     401-FURTRADE.BAS    51
     459-GOMOKO  .BAS    16
     478-GOLF    .BAS    70
     552-GUESS   .BAS     8
     562-GUNER   .BAS    14
     578-HAMURABI.BAS    46
NJ-2R005-HANGMAN .BAS    29
     063-HELLO   .BAS    28
     113-HEXAPAWN.BAS    47
     190-HI-LO   .BAS     9     SWTBASIC
     209-HI-Q    .BAS    27     S[illegible] BASIC
     265-HOCKEY  .BAS    66
     346-HURKLE  .BAS    11
     362-KINEMA  .BAS     7
     373-KING    .BAS    70
     451-LEM     .BAS    51

         SECTORS LEFT = 5


         CATALOG OF DRIVE #1    Vol 4
          NAME    TYPE  SIZE

     507-LETTER  .BAS     8
     517-LIFE    .BAS    15
     534-LIFE2   .BAS    19
     555-LITQUIZ .BAS    14
     570-LUNAR   .BAS    16
     587-MADLIB  .BAS    43
NJ-3 005-MASTMIND.BAS    37
     084-MATHDICE.BAS    11
     108-MUGWUMP .BAS    14
     135-NAME    .BAS     9
     152-NICOMACH.BAS     8
     167-NIM     .BAS    30
     215-NUMBER  .BAS     8
     230-ONECHECK.BAS    18
     258-ORBIT   .BAS    29
     297-PIZZA   .BAS    20
     325-POETRY  .BAS    11
     342-POKER   .BAS    67
     426-QUBIC   .BAS    55
     487-QUEEN   .BAS    32
     522-REVERSE .BAS    14
     539-ROCKET  .BAS    25
     564-ROCKSCIS.BAS     9
     576-ROULETTE.BAS    45
     621-RUSSROUL.BAS     6

         SECTORS LEFT = 8
```

Disk # 3

"101 Games" Vol 5

```
                    CATALOG OF DRIVE #1
                     NAME    TYPE   SIZE

NJ-3R  005- SALVO    .BAS     56
       117 - SINEWAVE.BAS      3
       126 - SLALOM  .BAS     37
       187 - SLOTS   .BAS     24
       226 - SPLAT   .BAS     39
       279 - STARS   .BAS     11
       296 - STOKMARK.BAS     45
       355 - SYNONYM .BAS     17
       377 - SKYDIVER.BAS     31
       416 - TARGET  .BAS     21
       440 - TICTAC1 .BAS     10
       454 - TICTAC2 .BAS     26
       489 - TOWER   .BAS     36
       529 - TRAIN   .BAS      6
       532 - TRAP    .BAS     12
       546 - TWEN3MAT.BAS     14
       562 - THREDPLT.BAS      3
       568 - WAR     .BAS     16
       585 - WORD    .BAS     16

            SECTORS LEFT = 148
```

"Misc" Vol 6

```
                    CATALOG OF DRIVE #1
                     NAME    TYPE   SIZE

Nd-4   [illegible] BIORHYT .BAS     33
       071 - NUMGAM  .BAS     16
       104 - HEX-DEC .BAS     15        JWT BASIC
       141 - INCTXM  .BAS    120
       312 - FINALBIO.BAS     21

            SECTORS LEFT = 371
```

Disk # 4

Programs from Publications

```
              CATALOG OF DRIVE #1
               NAME    TYPE  SIZE                           Vol 7
NJ-4
       341 - CONTENTS.TXT       5    Text Editor

       350 - WUMPUS   .BAS     57     SWTBASIC (all)
       435 - ROULETTE.BAS      30
       478 - MARKET   .BAS     62
       559 - CANNON   .BAS     20
       586 - TANK     .BAS     20
       614 - MUGWUMP  .BAS     16
       634 - SNARK    .BAS     13
NJ-4R  005 - ROADRACE.BAS      36
       099 - WGTCTRL  .BAS     37
       172 - BATNUM   .BAS     27
       226 - STARWARS.BAS     107
       381 - TIMEBOMB.BAS      20
       410 - SCRWORD  .BAS     11
       429 - ELIZA    .BAS     58
       501 - SAILORS  .BAS      7
       514 - HEXMATH  .BAS     11

              SECTORS LEFT = 33


              CATALOG OF DRIVE #1                           Vol 8
               NAME    TYPE  SIZE

       532 - INTRODUC.TXT       4   } Text Editor
       539 - CONTENTS.TXT      11   }      [illegible]
       550 - BASEBALL.BAS      45
       601 - ARTPRAC  .BAS     24
NJ-5   019 - WORLPR   .BAS     56 }
       005 - WORLPR   .TXT      5 }  Text Editor
       138 - BOWL     .TXT      1 }
       144 - BOWL     .BAS     32
       210 - LUNLAND  .BAS      9
       231 - SHOTSTR  .BAS     27
       284 - CHASE    .BAS     19
       320 - HORSE    .BAS     23
       362 - HI-LO    .BAS      5
       374 - CHKBAL   .BAS      9
       390 - RELADDR  .BAS      6
       402 - CSHFLOW  .BAS      7
       416 - MATHMIN  .BAS     16
       440 - FIBNUM   .BAS      2
       446 - SWRDSOR  .BAS     73
       538 - LUNLAN2  .BAS      8
       550 - UFO      .BAS     45
       600 - BOCCE    .BAS     33
NJ.5R  005 - DIETCAL  .TXT      1    Text Editor
       013 - DIETCAL  .BAS     57
       132 - GUNNER   .TXT      3    Text Editor
       141 - GUNNER   .BAS     24

              SECTORS LEFT = 67
```

Disk #5

Osborne Associates
Some Common Basic Programs

Vol #9

SWTBASIC (all)

```
          CATALOG OF DRIVE #1
           NAME    TYPE  SIZE

NJ-5R 191- INTRODUC.TXT     5  } Text Editor
      201- CONTENTS.TXT    15  }

      229- FUVALINV.BAS    12
      251- FUVAREDP.BAS    12
      273- REGDEPOS.BAS    11
      293- REWITHIN.BAS    12
      314- INITINV .BAS    11
      333- MININWIT.BAS    15
      357- NOMINRTI.BAS    13
      377- EFFINRTI.BAS    15
      400- ERNINTAB.BAS    25
      434- DEPRECRT.BAS     7
      446- DEPREAMT.BAS     9
      461- SALVVAL .BAS     8
      475- DISCOMPR.BAS     8
      487- PRINLOAN.BAS    10
      502- REGPAYLN.BAS    11
      519- LSTPAYLN.BAS    20
      544- REMBALLN.BAS    13
      561- ANINRTLN.BAS    16
      581- MORAMORT.BAS    29
      614- GCOMDEN .BAS     8
      625- PRIMFACT.BAS     4
NJ-6  005- AREAPOLY.BAS    10
      032- PARTTRIA.BAS    19
      078- ANAL2VEC.BAS     9
      103- OP2VECT .BAS     6
      120- RADTODEG.BAS     6
      136- DEGTORAD.BAS     6
      152- LININTER.BAS     9
      172- CURLININ.BAS    11
      195- SIMPRULE.BAS    15
      223- TRAPRULE.BAS    10
      242- GAUSQUAD.BAS    15
      269- DERIVATV.BAS     8
      276- ROOTQUAD.BAS     9
      302- NEWRTPOL.BAS    18
      332- RTOFPOLY.BAS    21
      365- TRIGPOLY.BAS    10
      381- SIMUEQU .BAS    11
      400- LINPROG .BAS    24

           SECTORS LEFT = 64
```

Disk #5   Osborne Associates
Some Common Basic Programs

Vol #10

Same as with Vol #9
SLUTBASIC (CRC)

```
          CATALOG OF DRIVE #1
           NAME    TYPE  SIZE

          INTRODUC.TXT      5

NJ-6 437- COORDCON.BAS     12
     456- CORDPLOT.BAS     28
     496- PLOTPOEQ.BAS     27
     532- MATRASSM.BAS     13
     551- MATRMULT.BAS     12
     568- MATRINVN.BAS     12
     585- PERMCOMB.BAS      6
     595- MANNWHIT.BAS     15
     615- MNVASTDV.BAS      8
     628- GEOMNDEV.BAS      5
NJ-6R005- BINDISTR.BAS      7
     028- BOISSOND.BAS      5
     047- NORMDIST.BAS      8
     069- CHISQRDS.BAS      8
     092- STUDTDIS.BAS      8
     114- STUDTDTS.BAS     18
     152- F-DISTRB.BAS     10
     174- LINCORCO.BAS      5
     188- LNREGRES.BAS     10
     211- NORDRREG.BAS     22
     253- GEOMREGR.BAS     11
     277- SYSTMREL.BAS      7
     292- AGREUPRO.BAS     12
     314- TAXDEPSC.BAS     15
     343- RECIPECT.BAS     10
     361- DAYOWEEK.BAS     10
     381- DABET2DA.BAS     20
     410- ANGLOMET.BAS     22
     441- ALPHABET.BAS     13

           SECTORS LEFT = 206
```

```
1457 RETURN
1458 SKIP #0,1
1459 RETURN
1460 GOSUB 1480
1464 SKIP #0,3
1467 PRINT #0,TAB(25);"COMPLIMENTS OF ---"
1469 PRINT #0,TAB(25);"LONG ENGINEERING CO. - WINSTON-SALEM, N.C."
1471 RETURN
1480 PRINT #0,TAB(10);
1481 PRINT #0,"THIS IS ONLY A DEMONSTRATION OF WHAT A COMPUTER CAN DO."
1482 PRINT #0,TAB(10);
1484 PRINT #0,"THIS COMPUTER IS USUALLY DOING ENGINEERING CALCULATIONS."
1486 RETURN
1490 PRINT "      END OF RUN MODE"
1500 END
```