Spirit of 99

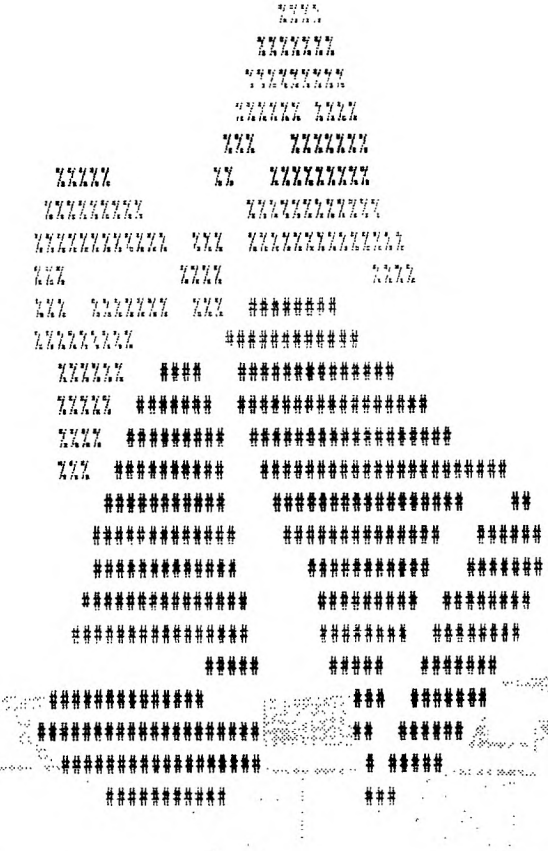
CENTRAL OHIO NINETY NINERS INC

THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED MONTHLY IN COLUMBUS OHIO

VOL 1 NO 12 DECEMBER 1983 $1.00

SEASON'S GREETINGS



Spirit of 99

VOL 1 NO 12 DECEMBER 1983

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO NINETY-NINERS

CENTRAL OHIO NINETY NINERS INC

Spirit of 99 is published monthly for Central Ohio ninety-niners inc. by Infoware Inc. and is the official news letter of (C.O.N.N.I.) USER GROUP. Editorial and Advertising address 1456 Grandview Avenue, Columbus Ohio, 43212. Subscription Address, 345 Glen Meadow Road, Dublin Ohio, 43017. Subscription Rate (USA) $13.oo One year (12 Issues), $24.oo Two years (24 Issues). Foreign Add $5.oo per year. Third class postage paid at Columbus, Ohio. CHANGE OF ADDRESS: Send both old and new Address to Subscription address above. WE assume no responsibility for manuscripts, programs, (tape or disk not accompanied by return postage. Letters to the Editor become the property of Spirit of 99. If published, We reserve the right to edit at our discretion. OPINIONS EXPRESSED HEREIN ARE THE AUTHORS AND ARE BASED ON VALID DOCUMENTABLE RESEARCH. THEY DO NOT NECESSARILY REFLECT THE OPINIONS OF THE PUBLISHER.

Central Ohio Ninety-Niners Inc. is a non-profit organization comprised of MEMBERS who own or use the TI994/A computer and it's related products and have paid a yearly membership fee of $15.oo and whose main objective is the exchange of EDUCATIONAL and SCIENTIFIC information for the purpose of computer literacy.

C.O.N.N.I. meetings are held on the Second Saturday of each month at the Martin Janis Senior Center on East Eleventh Avenue at the Ohio State fairgrounds. Meeting time is at 9:AM, Meetings are open to the public.
Membership dues ($15.oo) are payable yearly to C.O.N.N.I. and cover the immediate family of the member. (an application has been placed in this news letter for your convenience). Please address it to ADVERTISING ADDRESS ABOVE.

ADVERTISMENT: WE DO ACCEPT COMMERCIAL ADVERTISMENT AT THE FOLLOWING RATES;

BUSINESS CARD (2X3.5) $5.00/ISSUE
1/4 PAGE; $25.00
1/2 PAGE; $45.00
FULL PAGE; $75.00

WRITE THIS NEWSLETTER FOR OTHER SIZE ARRANGEMENTS. ALL ADS SHOULD BE SUBMITTED; CAMERA READY TO; ADVERTISING ADDRESS ABOVE, PAYMENT ENCLOSED. MEMBERS ADS ARE PUBLISHED AT NO CHARGE. (LIMIT 25 WORDS PLEASE).
IF YOU HAVE QUESTIONS, PLEASE CALL 486-7262 WEDNESDAYS ONLY 8AM-3PM. I WILL DO MY BEST TO HELP YOU.

PAT SATURN (ED)

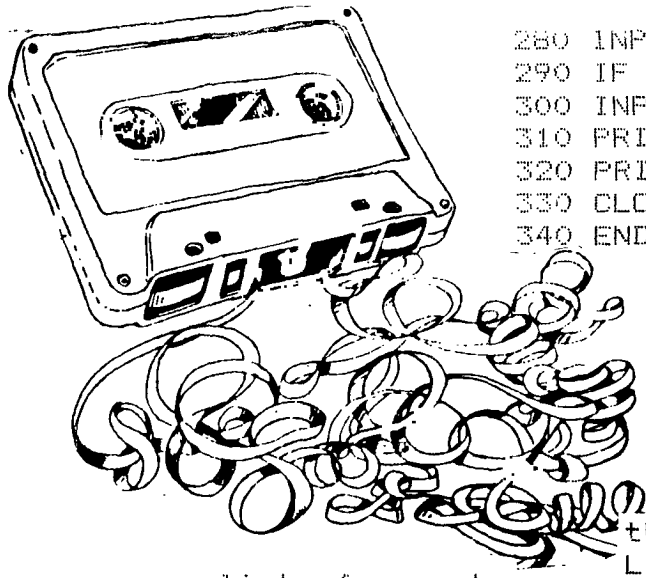
T. D. BELL

ON TAPE

Last time, we left off with SAVEing on tape. Retrieve v. get back, that is exactly what comes next. Old CS1 is covered very well in the basic manual, so we will only touch briefly on this.

As with SAVE you get screen prompts to help you along. However the error messages are quite cryptic to say the least. "NO DATA FOUND", means the tape machine volume control has to be adjusted to a higher level. IF it has a tone control, start by turning it to the high end of the sound frequency (treble).

"ERROR IN DATA" is the other message you will get that has to do with the volume control, it generaly means volume is too high and must be turned down. When either of these error messages are received, simply press SHIFT E. You will get another message, (I/O ERROR). Ignore it, follow the prompts, adjust the volume control on your tape machine, and try again. After a few trys, you should reach a place on the volume control that is compatable with your computer. If you do not the tape player is most likley not compatable with the computer. Try another brand. Ask some of the old timers what they are using, or send a self address ed stamped envelope to T.D.BELL c/o this column and I will get you a list of compatable machines.



AND NOW
AS PROMISED
DATA FILES ON TAPE

Type in the following program and save it to CS1.

```
10 CALL CLEAR
20 CALL SCREEN (8)
30 PRINT "T.D.BELLS CASSETTE ADDRESS":"BOOK ON TAPE ":"PRESS 1 FOR FILING "::"
    2 FOR RETRIEVE":::::
40 CALL KEY(0,K,S)
50 IF S=0 THEN 40
60 IF K=ASC("2")THEN 260
70 IF K=ASC("1")THEN 90
80 GOTO 40
90 REM  CASSETTE ADDRESS BOOK FILE DEMO
100 OPEN #1:"CS1",OUTPUT,FIXED
110 CALL CLEAR
120 PRINT "NAME PLEASE? "::"TYPE <END> TO STOP"::
130 INPUT A$
140 IF A$="END" THEN 230
150 PRINT "ADDRESS OF ";A$::
160 INPUT B$
170 PRINT "CITY AND ZIP "::
180 INPUT C$
190 PRINT #1:A$
200 PRINT #1:B$
210 PRINT #1:C$
220 GOTO 110
230 CLOSE #1
240 END
250 REM  CASSETTE ADDRESS BOOK RETRIEVE FILE DEMO
260 OPEN #1:"CS1",INPUT ,FIXED
270 CALL CLEAR
280 INPUT #1:A$
290 IF A$="END" THEN 330
300 INPUT #1:B$,C$
310 PRINT A$:B$:C$
320 PRINT " "
330 CLOSE #1
340 END
```

This short program will ask for information, Name, Address, and zip code of your address book, A$, B$, and C$ LINES 130, 160 and 180 to the file that was opened in LINE 100. The file is a place in memory that we wish to store our information for later, or remove information from for imediate use. This file must have a number (from 1 to 255) We will use #1 to keep it simple. Next is the word OUTPUT, which is to say we wish to output that which is in the file to storage device CS1. As would naturaly follow INPUT would be to bring the information in from the cassette to the file. This information may be put into the file or removed in either of two forms; DISPLAY, which is ASCII form, or the way you see it on the screen; or INTERNAL, which is machine code non printable characters which also uses less space in memory. Data in this form can not be read by people. The other difference in these two forms is, with display type records the computer has to translate from ASCII to machine code and then back again. This obviously slows down a basic program. If this specification is omitted the compu-

ter assumes you want DISPLAY.

RECORD TYPE: do you want your records all to be the same length or will they vary? If you specify FIXED, you then have a choice of 64, 128, or 196 length, (if you do not specify the computer assumes 64). With VARI ABLE you may have the same choices allow ing you less than but no more than the speci fied length for the record. We have chosen FIXED and defaulted to 64 as our record length. Also the compu ter will pad all of our records with spaces to in sure that they are 64 characters each. All cassette files are sequential.

Files can be SAVED to either CS1, or CS2, however they can only be retrieved from CS1. Most likley because CS2 has no provision for listening. (no wire connected between cassette speaker and the computer).

The seconed part of the program, LINE 250, will get your file back from the cassette whenever you wish. As I mentioned earlier the program is simple so as to make it under standable. It has room for many improvements, suggestions would be; DIMention variables A$, B$ and C$ for a quantity of more than 10. Variables will default to 10 unless they are DIMentionted for more. Put in a loop to get back to the menu and end your program in an orderly fashion. There are many possiblities for wistles and BELLs...

Until next year,

HAPPY HOLIDAYS
AND
HAPPY COMPUTING.

T.D.BELL

TENDERFOOT BASIC

BY NIRAJ N. SHAH

This month I am going to show how to convert Joystick based programs to work for the Keyboard.

First lets work on the Joystick to Keyboard conversion routine. The first thing that is needed is to make the computer recognize that we are using the Keyboard instead of the Joystick. This means that the CALL KEY statement has to be used and the CALL JOYST statement must be ignored. To make the computer ignore the CALL JOYST we must use an IF THEN ELSE construct to decide if the user wants to use the Keyboard or the Joystick. So, why not put a question at the beginning of the program in the form of the INPUT statement asking the user if he wants to use the Joystick or the Keyboard?! Then you have to test the user's response to the question. Here is how to do that:

```
100 REM BEGINNING OF THE
PROGRAM
110 REM ASK THE QUESTION
120 INPUT "USING THE KEYBOARD?
(Y/N) ";REPLY$
130 IF REPLY$="Y" THEN 140 ELSE
180
140 REM GOTO KEYBOARD ROUTINE
150 GOSUB 500
160 REM AVOID THE JOYSTICK PART
170 GOTO 190
180 CALL JOYST(1,X,Y)
    :
    :
500 REM THE KEYBOARD ROUTINE
    :
    :
900 REM END OF KEYBOARD ROUTINE
910 RETURN
```

The above program asks the user in LINE 120 if he is going to be using the Keyboard via the use of the INPUT statement. Lets assume if the user does not give the response of "Y" then he does NOT want to use the Keyboard. So, when the user responds to the question put forth in LINE 120 his response will be stored in the variable REPLY$. Then LINE 130 tests the user's reply via the use of the IF THEN ELSE statement. If the user's response, which is stored in the variable, (REPLY$), is "Y" that means he wants to use the Keyboard. So LINE 130 will recognize that (REPLY$) is equal to "Y" and will branch to line 140. Then LINE 140 sends the computer to the part of the program that scans the Keyboard via the use of the GOSUB statement. Line 140 sends the computer off to LINE 500 with the agreement that he will come back to LINE 150 when he encounters a RETURN statement. That is why LINE 910 has a RETURN statement. This will make the computer return to LINE 150 after he has finished executing the Keyboard routine. When the computer does come back to LINE 150 he sees the GOTO 190 statement. So, he jumps to line 190 and continues to execute the program from that point (LINE 190). The reason for LINE 150 is to avoid the Joystick part of the program. After all, the objective of the Keyboard routine was to substitute for the Joystick routine. This way one does not have to delete the Joystick routine to have a Keyboard routine. The user is merely given the option between using the Joystick or Keyboard.

Now lets see what happens if the user's response to the INPUT statement in LINE 120 is NOT "Y". So, the variable, (REPLY$) is not equal to "Y" thus the IF THEN ELSE statement in LINE 130 will prove to be false. Because (REPLY$) is not equal to "Y" the ELSE part in LINE 130 will be executed. This means that the computer is directed to go to the Joystick part of the routine which begins at LINE 180. Once the computer branches to LINE 180 it begins executing from that point.

Now suppose that the computer has finished executing either the Joystick or the Keyboard routine and is located somewhere around LINE 300. What is going to keep the computer from falling into the Keyboard routine which begins at LINE 500?! Nothing!! Unless you put an END or a STOP statement before LINE 500. Just make sure that the line number of the END statement is below 500 but not too far below it. Lets say that the END statement is in LINE 400. Now as long as the computer is not allowed to venture into the region between LINEs 400 and 500 the computer will not accidentally fall into the Keyboard routine.

Now that you have seen how to test the user's response to the Joystick or Keyboard question lets get to the good part. Lets fill in the spaces between Lines 500 and 910 for the Keyboard routine.

The first thing that has to be decided is which keys correspond to the direction of motion. Lets adopt the following convention:

JOYSTICK-TO-KEYBOARD	JOY STICK	
USING THE KEYBOARD	X	Y
[E]=UP	0	+4
[X]=DOWN	0	-4
[S]=LEFT	-4	0
[D]=RIGHT	+4	0
[R]=DIAGONALY UP-RIGHT	+4	+4
[W]=DIAGONALY UP-LEFT	-4	+4
[Z]=DIAGONALY DOWN-LEFT	-4	-4
[C]=DIAGONALY DOWN-RIGHT	+4	-4

The above table shows the KEYS in the left column that the user can use to move in the indicated directions. The right column, JOYSTICK, shows the corresponding parameters for the CALL JOYST subprogram to move in the indicated directions.

Because the program that you are trying to convert to Joystick/Keyboard use already has the program code that uses the Joystick parameters you do not have to worry about them too much. Your job is to write the code that scans the Keyboard and interprets the user's response so that the built-in Joystick code can use that information. Here is how to do it using the above table as a guide:

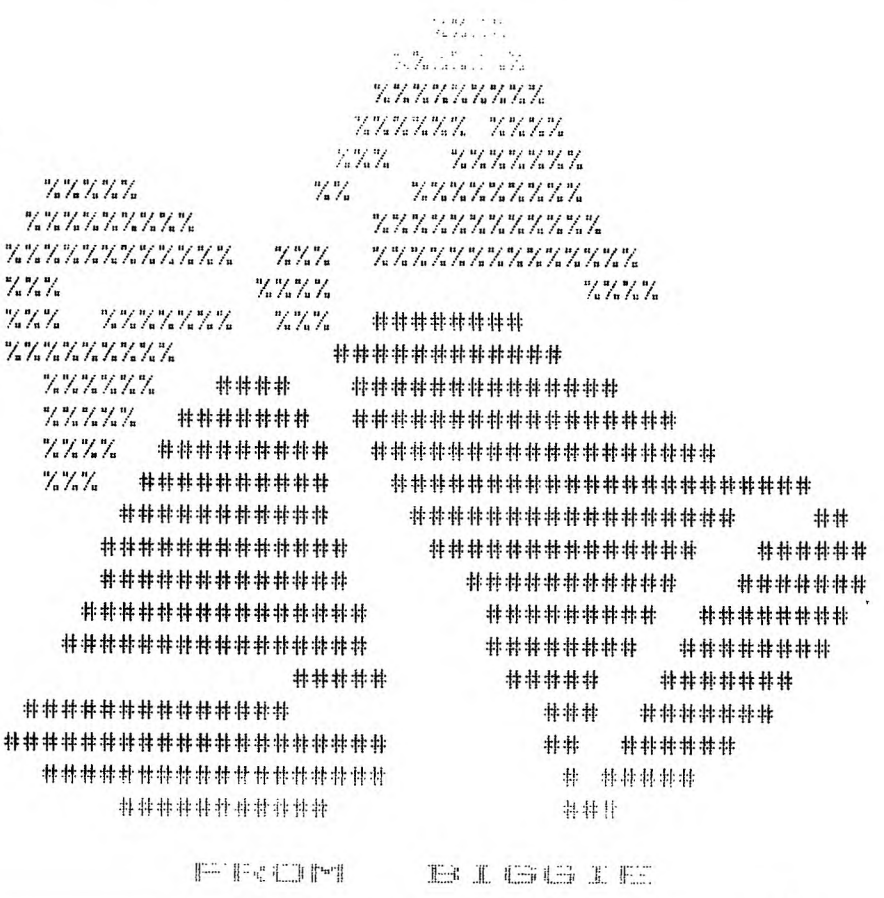
```
500 REM THE KEYBOARD ROUTINE
510 REM SCAN THE KEYBOARD
520 CALL KEY(0,KEY,ST)
530 IF ST=0 THEN 531 ELSE 540
531 X=0
532 Y=0
540 REM A KEY WAS PRESSED
550 IF KEY=ASC("E") THEN 560
ELSE 580
560 X=0
570 Y=4
580 IF KEY=ASC("X") THEN 590
ELSE 700
590 X=0
600 Y=-4
700 IF KEY=ASC("S") THEN 710
ELSE 730
710 X=-4
720 Y=0
730 IF KEY=ASC("D") THEN 740
```

```
100 REM DECIMAL TO HEXADECIMAL
110 REM CONVERSION FOR MEMORY
120 REM MAPS
130 REM  SIEVE 04/15 (1983)
135 REM MODIFIED BY ROGER WILLS (1983)
140 CALL CLEAR
145 PRINT "DECIMAL TO HEXADECIMAL"
147 PRINT "CONVERSION INCLUDING"
149 PRINT " NEGATIVE NUMBERS."
150 PRINT
200 H=16
210 PRINT "ENTER THE DECIMAL NUMBER"
215 PRINT "YOU WANT TO CONVERT"
220 INPUT N
222 IF SGN(N)=1 THEN 230
224 N=N+65536
230 A$=""
240 NP$=STR$(N)
250 J=1
260 FOR I=1 TO 10
270 K(I)=J*H
280 J=K(I)
290 NEXT I
300 FOR I=10 TO 1 STEP -1
310 IF K(I)<N THEN 340
320 L(I)=0
330 GO TO 360
340 L(I)=INT(N/K(I))
350 N=N-(L(I)*K(I))
360 NEXT I
370 L(0)=N
380 FOR I=10 TO 0 STEP -1
390 IF L(I)<>15 THEN 410
400 A$=A$&"F"
410 IF L(I)<>14 THEN 430
420 A$=A$&"E"
430 IF L(I)<>13 THEN 450
440 A$=A$&"D"
450 IF L(I)<>12 THEN 470
460 A$=A$&"C"
470 IF L(I)<>11 THEN 490
480 A$=A$&"B"
490 IF L(I)<>10 THEN 510
500 A$=A$&"A"
510 IF L(I)>=10 THEN 530
520 A$=A$&STR$(L(I))
530 NEXT I
535 X$=SEG$(A$,8,4)
540 PRINT X$
550 PRINT
560 GO TO 210
570 END
```

```
10 REM HEXADECIMAL TO DECIMAL
11 REM  CONVERSION FOR MEMORY MAPS
12 REM BY ROGER WILLS(1983)
100 CALL CLEAR
105 PRINT "HEXADECIMAL TO DECIMAL"
107 PRINT "CONVERSION FOR MEMORY MAPS"
```

```
110 INPUT "HEX NUMBER ";H$
120 N=1
130 DEC=0
140 FOR J=1 TO LEN(H$)
150 A$=SEG$(H$,LEN(H$)-J+1,1)
160 IF ASC(A$)>58 THEN 190
170 HT=VAL(A$)
180 GO TO 200
190 HT=ASC(A$)-55
200 DEC=DEC+N*HT
210 N=N*16
220 NEXT J
225 IF (DEC)<32768 THEN 230
226 DECX=-(65536-DEC)
227 PRINT "DECIMAL=";DECX
228 PRINT
229 GO TO 110
230 PRINT "DECIMAL=";DEC
235 PRINT
240 GO TO 110
250 END
```

SEASON'S GREETINGS



FROM BIGGIE

```
ELSE 770
750 X=4
760 Y=0
770 IF KEY=ASC("R") THEN 780
ELSE 800
780 X=4
790 Y=R
800 IF KEY=ASC("W") THEN 810
ELSE 830
810 X=-4
820 Y=4
830 IF KEY=ASC("Z") THEN 840
ELSE 860
840 X=-4
850 Y=-4
860 IF KEY=ASC("C") THEN 870
ELSE 900
870 X=4
880 Y=-4
900 REM END OF KEYBOARD ROUTINE
910 RETURN
```

The above program works as an all-purpose subroutine to convert a Joystick based program to a Joystick and a Keyboard based program. I am not going to explain how this subroutine works because it is very similar to the way the main program, shown above, works. What I will explain is the ASC("E") statement.

The ASC statement converts the alphanumeric character enclosed within the quotes into a code that is understood by the computer. The name of this code is called ASCII code. For example, the dollar($) sign's ASCII code is 36.

For all you lucky ones who have EXTENDED BASIC can shorten the above Keyboard subroutine into even shorter segments. Here is a short example:

```
500 REM FOR EXTENDED BASIC ONLY
510 REM KEYBOARD ROUTINE
520 CALL KEY(0,KEY,ST):: IF
ST=0 THEN X,Y=0
530 IF KEY=ASC("E") THEN X=0 ::
Y=4 ELSE IF KEY=ASC("X") THEN
X=0 :: Y=-4
540 IF KEY=ASC("S") THEN X=-4
:: Y=0 ELSE IF KEY=ASC("D")
THEN X=4 :: Y=0 :: RETURN
```

Here is something for you to mull over until next month. In the regular BASIC version of the Keyboard routine there is a way you can speed up the response of the program with respect to the Keyboard. The solution is to add something before each IF THEN ELSE construct. Think about it and if you can not figure it out then TOUGH LUCK!!

ASSEMBLER

With assembly language a mistake will cause the machine to lock up and the only way to break the program is to shut off the computer. Always remember to make a copy of your program before you try to run it. If not, I hope you like to type, because you will lose the program when you must turn off the computer when it locks up.

We will start with a little program to display something on the screen. This program is written for the Editor Assembler, but can be entered with the MINI-MEM with minor changes.

In order to print something to the video RAM (VDP RAM), register 0 (R0) must contain the address where the information will be stored; register 1 (R1) must have the address of the information to be moved; and if more than one byte register 2 (R2) must have the byte count.

```
        REF   VSBW,KSCAN
        DEF   START
STATUS  EQU   >837C
START   LI    R0,>0022
        LI    R1,>4200
        BLWP@VSBW
SCAN    BLWP@KSCAN
        MOVB@STATUS,R0
        JEQ   SCAN
        CLR   R0
        MOVB  R0,@STATUS
        B     *R11
        END
```

Here is a line by line explaination of the program.

1 REF allows the program to link to some built in utility programs. In this program I am using two utilities VSBW (VDP single byte write) and KSCAN (scans the keyboard).

2 DEF is the starting point of the program. The start of the program can be anywhere in the program and can have almost any name.

3 EQU gives a name (STATUS) to a location (>837C) or area.

4 START is where the program actually starts. LI R0,>018E Loads Immediately register zero with hex 18E.

5 LI R1,>4200 Loads Immediately register one with the hex for "B" (42) and fills the rest with zeros.

6 BLWP@VSBW Branches and Loads the Work Pointer with the address found at VSBW and starts executing from that address.

7 SCAN BLWP@KSCAN The program runs so fast that I had to give you some time to see what it did. So I branched and loaded the address of the keyboard scan utility.

8 MOVB@STATUS,R0 MOVes the Byte at location STATUS into register zero. When the computer moves, adds, subtracts, and some other things the appropriate bits in R15 (status register).

9 JEQ SCAN jumps to scan if the equal bit is set.

10 CLR R0 CLears register zero.

11 MOVB R0,@STATUS clears location STATUS so that no error is indicated.

12B *R11 Branches to the instruction at the address in register eleven which was loaded there when the program started.

13 END does not need to be there but I don't like to see any error messages when I assemble a program.

This program does the same thing as listing 2 in Roger Wills artical called POKE'N AROUND THE MINIMEM.

Page 403 in the E/A manual gives the various memory uses and their range in the Video Display Processor ram. If you wanted to move to some other location on the screen change the value in R0. To change the character displayed change the first two byte in the word in R1.

If more than one byte is to moved at once, include in the REF line VMBW (Vidio Multiple Byte Write). List the information in a DATA line with a name and Load Immediately R1 with the name of the DATA line. Load Immediate ly R2 with the word count (two bytes=one word). Branch and Load Work Pointer at VMBW.

Problems to work on: Put the same information into several consecutive locations; Print the alphabet on the screen; Change the color of the screen.

Enough for now. Remember to see me or give me a call if you are interested in attending assembly language group meetings. Please see related artical for information. MIKE

THE SKY IS NOT FALLING

Excerpt from A9CUG CALL newsletter. Thomas H. Boisseau.

In a recent conversation with Ed Wiest, you remember Ed, he stated that the exchange centers would remain open to service 4A owners. We are also informed that all software that had been announced for 1983 will be produced and distributed, and that all current soft-ware packages will continue to be produced. TI IS CONSIDERING THE POSSIBILITY OF FURTHER SUPPORTING THE 994/A BY CONTINUING TO INTRODUCE NEW SOFTWARE TITLES.

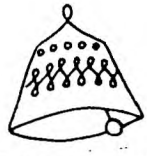
On the hardware end, Ed stated that TI will continue to produce peripherals only untill ALL PRESENT AND BACK ORDERS ARE FILLED.

The Consoles have already ceased production, however cnsidering the present warehouse inventory, he estimates they will remain on the market until second quarter of 84.

<<<<PIO PIN OUT FOR TI>>>>

Connect TI pin 1 to centronics connector pin 1 / TI pins 2-9 to centronics pins 2-9 / TI pin 10 to centronics pin 11 / TI pin 11 to centronics pin 29 / TI pin 16 to centronics pin 16.

These should work for most printers TI Epson, Epson, etc. If you have another application please contact this newsletter in writing, also if you need help with a printer. (Stolen from A9CUG newsletter, November).



A CHRISTMAS TREAT FROM MILLERS GRAPHICS

DSK2.JBELLS

```
1 ! <<< JINGLE BELLS >>>
                by
          Craig Miller
        MILLERS GRAPHICS

2 ! Have a Happy Holiday
 and a Prosperous New Year.

10 DATA 250,494,250,494,500,
494,250,494,250,494,500,494,
250,494,250,587,250,392,250,
440,1000,494,250,523,250,523

20 DATA 250,523,250,523,250,
523,250,494,250,494,250,494,
250,494,250,440,250,440,250,
494,500,440,500,587,250,494

30 DATA 250,494,500,494,250,
494,250,494,500,494,250,494,
250,587,250,392,250,440,1000
,494,250,523,250,523,250,523

40 DATA 250,523,250,523,250,
494,250,494,250,494,250,587,
250,587,250,523,250,440,1000
,392

50 RESTORE :: FOR T=1 TO 49
:: READ A,B :: CALL SOUND(A,
B,4):: NEXT T

60 RESTORE :: FOR T=1 TO 49
:: READ A,B :: CALL SOUND(A,
B,2,B/2,6):: NEXT T

70 RESTORE :: FOR T=1 TO 49
:: READ A,B :: CALL SOUND(A,
B,2,B*2,4,B/2,6):: NEXT T

80 RESTORE :: FOR T=1 TO 49
:: READ A,B :: CALL SOUND(A,
B,2,B+5,4,B/2,6):: NEXT T

82 ! To change the speed of
the song, multiplying  A in
the sound statement by .9 or
 .8
ie CALL SOUND(A*.9,B,2  etc.
```

Just a Note from MILLERS GRAPHICS

Many people have written or called us to see if we are still going to support the TI home computer, and the answer is definitely YES. We are also going to publish the book on Extended BASIC programming that we announced in our newsletter and we currently have some other exciting new products on the drawing board.

Sincerely,

Craig Miller
Millers Graphics

press sez

MEMORY MAPS by ROGER WILLS

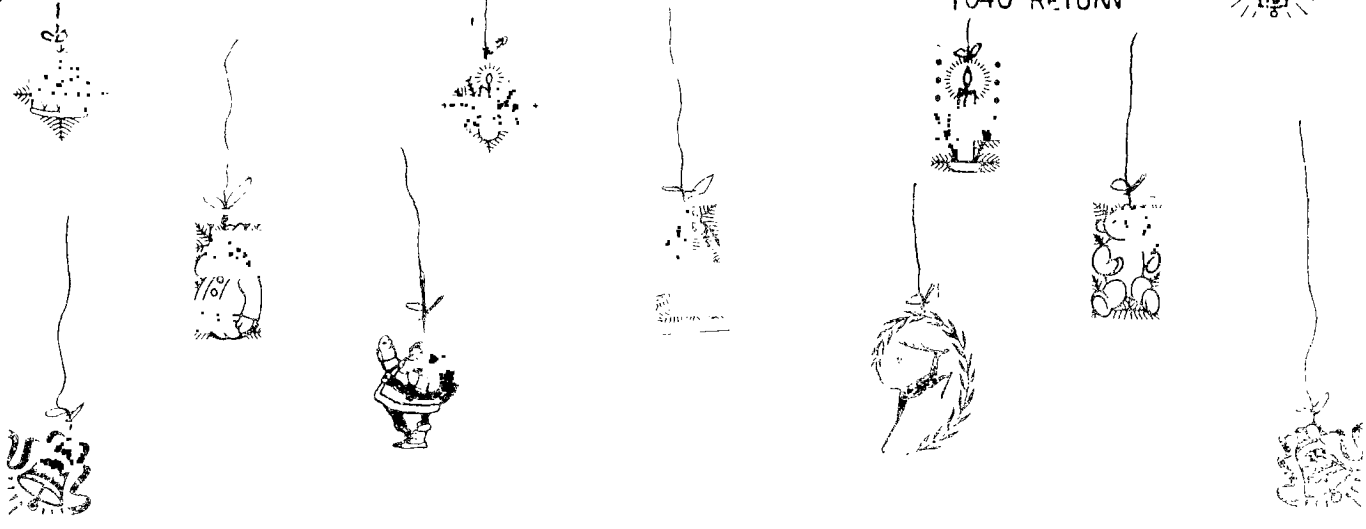
Information is stored in a number of areas in the computer's memory,principally in RAM and ROM.The smallest unit is called an address or a location.The term MEMORY MAP is used to describe what functions these areas perform or what UTILITY programs they contain.For example addresses >0000 to >1FFF contain the console ROM(see MINIMEM BOOKLET).
If you are involved in ASSEMBLY or just PEEKING and POKING you will find these two listings useful for all those conversions since they handle the negative numbers found in the MEMORY MAPS.

RUMOURS

With TEXAS INSTRUMENTS announcement to stop manufacturing the 99/4A computer there are alot of rumours around. Since a substantial number of rumours have a habit of becoming a reality it is worthwhile to summarise what these are:
1)Two major companies are negotiating to continue production of the 99/4A computer.
2)There is an advanced version of the 99/8 computer being developed.
3)Increased amounts of SOFTWARE by large and small companies will be available for the 99/4A.
The general reaction of the marketplace has been extremely positive,and I look forward to interesting announcements in the next few months.

ROGER WILLS

```
100 CALL CLEAR
110 PRINT "  THE TIGERCUB'S
CHRISTMAS"::TAB(11);"ORNAMEN
T"::"BY JIM PETERSON"::
120 PRINT " USE ARROW KEYS A
ND W,R,Z":"AND C KEYS TO MOV
E CURSOR";":"THEN PRESS ANY O
THER KEY TO"
130 PRINT "PRINT DESIGN, OR
ENTER TO":"SKIP. USE BOTH SH
IFTED AND":"UNSHIFTED LETTER
S AND PUNCT-"
140 PRINT "UATION, ALSO TYPE
 ANY LETTER":"WITH CTRL HELD
DOWN."
150 PRINT " USE FCTN 1 TO CH
ANGE PAT-":"TERN OF LAST CHA
RACTER PRIN-";"TED, OR FCTN
2 TO CHANGE ITS"
160 PRINT "COLOR, OR FCTN 3
TO CHANGE":"ALL THE COLORS,
OR FCTN 7":"TO SLOWLY CHANGE
 ALL THE"
170 PRINT "PATTERNS - BUT ST
AY AWAY":"FROM FCTN 4 AND FC
TN = 1":"          PRESS ANY
KEY"
180 CALL KEY(0,K,ST)
190 IF ST=0 THEN 180
200 DIM A$(16)
210 CALL CLEAR
220 FOR SET=1 TO 16
230 CALL COLOR(SET,5,16)
240 NEXT SET
250 CALL VCHAR(1,31,1,96)
260 CALL SCREEN(2)
270 FOR CH=33 TO 159
280 PRINT CHR$(CH);" ";
290 Z=Z+1
300 IF Z/14<>INT(Z/14)THEN 3
20
310 PRINT ::
320 NEXT CH
330 GOSUB 630
340 GOSUB 940
350 CALL VCHAR(1,3,32,672)
360 CALL VCHAR(1,29,1,192)
370 R=6
380 C=11
390 CALL KEY(3,K,ST)
400 IF ST=0 THEN 390
410 IF K<68 THEN 820
420 ON POS("WERDCXZS",CHR$(K
),1)+1 GOTO 390,490,470,460,
440,430,530,520,500
430 R=R+ABS(R<12)
440 C=C+ABS(C<16)
450 GOTO 540
460 C=C+ABS(C<16)
470 R=R-ABS(R>1)
480 GOTO 540
490 R=R-ABS(R>1)
500 C=C-ABS(C>5)
510 GOTO 540
520 C=C-ABS(C>5)
530 R=R+ABS(R<12)
540 CALL HCHAR(R,C,42)
550 CALL KEY(5,CH,ST)
560 IF ST=0 THEN 550
570 IF (CH=87)+(CH=69)+(CH=8
2)+(CH=83)+(CH=68)+(CH=90)+(
CH=88)+(CH=67)THEN 550
580 CALL HCHAR(R,C,CH)
590 CALL HCHAR(25-R,C,CH)
600 CALL HCHAR(R,33-C,CH)
610 CALL HCHAR(25-R,33-C,CH)
620 GOTO 390
630 DATA 00,18,24,3C,42,5A,6
6,7E,81,99,A5,BD,C3,DB,E7,FF
640 FOR J=1 TO 16
650 READ A$(J)
660 NEXT J
670 FOR CH=33 TO 159
680 GOSUB 700
690 GOTO 800
700 FOR L=1 TO 4
710 RANDOMIZE
720 X=INT(16*RND+1)
730 B$=B$&A$(X)
740 C$=A$(X)&C$
750 NEXT L
760 CALL CHAR(CH,B$&C$)
770 B$=NUL$
780 C$=NUL$
790 RETURN
800 NEXT CH
810 RETURN
820 ON K+1 GOTO 390,830,390,
850,880,390,390,920
830 GOSUB 670
840 GOTO 390
850 IF (CH<33)+(CH>159)THEN
390
860 GOSUB 700
870 GOTO 390
880 SET=INT(CH/8)-3
890 IF SET<2 THEN 390
900 GOSUB 980
910 GOTO 390
920 GOSUB 940
930 GOTO 390
940 FOR SET=2 TO 16
950 RANDOMIZE
960 GOSUB 980
970 GOTO 1030
980 X=INT(15*RND+2)
990 Y=INT(15*RND+2)
1000 IF Y=X THEN 990
1010 CALL COLOR(SET,X,Y)
1020 RETURN
1030 NEXT SET
1040 RETURN
```



SEE US AT THE CHRISTMAS FAIR

Software in Xbasic

SPRITEGET: Draw your sprites
with ease.$12

POLAR GRAPHICS: Learn how to
draw curved patterns

POLKADOT: Make multicolored pictures
with ten different colors.$7.00

JEDI MATH: Improve your math skills,
CHALLENGE of powers.$18.00

TEACHERS' AID: The easy way to handle
each student's score
in every subject.$19.00

PORTFOLIO MANAGER: Manages up to 50 stocks

,total portfolio,total $s
invested.$25.00

etc.$25.00

STOCKPLOT: Makes stock price movement easier
to evaluate.Tracks price movements
over days,weeks,months or years with
highs,lows and closing prices.$25

ohio residents add 5.5% sales tax

$1.50 shippinghandling

Software by TEXAS INSTRUMENTS

SOFTWARE CAROUSEL

AMLIST INFOWARE

90 ¢ each
10 min order

cassetts

$1.20 W/BOX

Voice synchronised demo

GRAPHICS CODE GENERATOR

You will LOVE IT.SEE IT at
THE XMAS FAIR

Intelligent Software by INFOWARE INC

SEND CHECK TO:

INFOWARE INC
345 Glen Meadow Court
Dublin,Ohio 43017

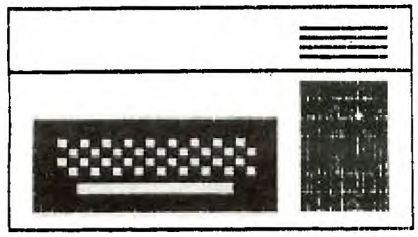
Tel:486-7262 or 889-9011

Spirit of 99

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO NINETY-NINERS

PUBLISHED MONTHLY IN COLUMBUS OHIO

CENTRAL OHIO



NINETY NINERS INC

REASON FOR BUYING

( ) ENTERTAINMENT
( ) EDUCATION
( ) COMPUTER LITERACY
( ) JOB RELATED
( ) OTHER (_____)

99 OWNERS OR BUYERS

C.O.N.N.I. WANTS YOU

Looking for HELP? Other Users? Discounts? Access to a growing software library? Do you need technical advice? Can you give technical advice? Have a talent with computers you'd like to share with other users?

WELL STOP LOOKING!!!

Stop sitting up late trying to figure out what "BAD VALUE" means, call someone who does, or someone who doesn't, or just call someone to be confused with you.

ALL THIS AND MORE!!!!

We are the fastest growing TI user's group in the world. We are located in Central Ohio (address below) We will do our best to help you find the answers to your many questions and problems about your TI computer. We also have a helpful newsletter, (The Spirit of 99) recognized by TI as one of the best.

Our MEETING place is the Martin Janis Senior Center, Located on East 11th Avenue (fair grounds) in Columbus (just west of I 71)
Our Meetings are at 9AM on the SECOND SATURDAY of EACH MONTH, and open to the public. (Time or Date changes will be noted in the newsletter). Write to C.O.N.N.I. at 1456 GRANDVIEW AVENUE COLUMBUS OHIO 43212. If you have a question about us, call BIGGIE at (614) 486-7262 MONDAY-WEDNESDAY 8AM-2PM ONLY, or come to a meeting.

If you would like to join our users group the dues are $15.00 per year (single or family). An application form is attached. If you would rather not join but would like to receive our newsletter, Subscription is $12.00 PER YEAR. (Members receive the newsletter FREE). ——→

*** MEMBERSHIP APPLICATION ***

NAME _____ AGE _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
AREA CODE HOME PHONE _____ BUSINESS PHONE _____ EXT# _____
WHAT IS YOUR PROFESSION/VOCATION _____
HOW LONG HAVE YOU OWNED YOUR COMPUTER _____
DATE OF APPLICATION _____ ACCEPTED BY _____

Bulk Rate
U.S. Postage
PAID
COLUMBUS 43212
Permit No. 1945

EDMONTON USERS GROUP
P O BOX 11983
EDMONTON ALBERTA T5J3L1

FROM C.O.N.N.I.