MID ILLINOIS COMPUTER RESOURCE ORGANIZATION
P. O. BOX 766 BLOOMINGTON, IL
61701-766

# BLOOMINGTON - NORMAL MICRO NEWSLETTER

MAY 1985, VOL. 3, No. 5

Welcome to the age of the computer in the home! Future meetings will be held the third THURSDAY of each month at 7 P.M. at the IAA Building in Bloomington . Future dates include July 18, August 15 and September 19.

*********

The program for July will consist of everyone attending to demo a cassette program or a cartridge. A cassette recorder and Extended Basic will be available. There will not be a disk drive to use. I hope everyone brings something to share. This will give some of our members a chance to demo a program for their first time.

## ***** PRESIDENTS NOTES *****

Although you can't tell it by our group (we only had 11 members at our June meeting), interest in the TI appears to be getting stronger. My favorite TI related publication MICROpendium is now 48 pages long. Home Computer Magazine is being published more frequently. New software and hardware is being released all the time. Many TI owners have been waiting for some upgraded computer. I have not heard if the rumored "99/8" was shown at the Consumer Electronics Show, but the latest issue of MICROpendium has an article about a TI-compatible computer and a line of peripheral systems. The new hardware will be marketed by Tex-Comp of Granada Hills, CA. There will be 3 peripheral systems; TC-1, TC-2, and TC-3. They all will have an enclosure with 2 double-side, double-density half-height drives. The top-of-the line system, TC-3, will have also have 128k RAM, speech synthesizer, and built-in CorComp 9900 system. It will retail for $875. The new computer, TC-99/9, will have the TC-3 system, 80 column TI processor board, and use the new Wico keyboard. With a color monitor, it will retail for $1300-$1500. Hopefully, we will more new products like this.

Brian McFeeters

Sam Shank and Aubrey Johnson announced at the June meeting that the programs we got from the Champaign Users Group were available to checkout. Our club now has 29 disks of new programs. However, none of the disks have been backed-up. If you check any disks, please get a blank disk from Sam or Aubrey and make a back-up. Also, a REM statement needs to be added to the begining of each program. It should read: ACQUIRED FROM WW99ERS. Thanks again Sam and Aubrey for all of your work.

**********************************************************************

Our club added two more members last month. They are Tara Gruel of Normal and Larry Scovel of Bloomington. Welcome to the club! We now have 26 members.

**********************************************************************

Thanks again to Jim Peterson for his TIPS FROM THE TIGERCUB.

**********************************************************************

The following program has been in many newsletters. One gave credit to T.Atkinson. The program uses a series of CALL LOADS to redefine the shape of the cursor. The CALL LOADS use memory space that EXTENDED BASIC does not use. You can run the program with or without line numbers. Load this program first and then run any EXTENDED BASIC program. Any time the cursor appears it will be in the shape of Texas. The program requires EXTENDED BASIC and 32K memory expansion.

```
100 REM CURSOR REDEFINER
110 CALL INIT
120 CALL LOAD(8196,63,248)
130 CALL LOAD(16376,76,85,82,83,79,82,48,8)
140 CALL LOAD(12288,48,48,63,255,254,124,24,12)
150 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,36,4,91)
160 CALL LINK("CURSOR") :: END
```

The new cursor shape is created by line 140, in which the eight values following the memory location equal the decimal equivalent of the hex values in a CALL CHAR statement. You can change the shape of the cursor by changing the eight numbers following the number 12288 in line 140. If line 140 is removed, the cursor now becomes a flat underline. If anyone comes up with another shape, we will publish it in our newsletter.

**********************************************************************

From Donna Griffin - Hoosier User Group

For use with AMAZING cartridge: if you hit FCTN 3(erase) , you will get the time clock to appear while the game is playing.

The following program ,NUMTALK, was in the MAY85 MANNERS NEWSLETTER of the Mid Atlantic Ninety Nine'ers. The article about NUMTALK was written by Maurice E.T. Swinnen.

## NUMTALK (for X-basic only)

CALL SAY statements lack the ability to pronounce a number between 10 and 999 correctly. For example, CALL SAY("123") will result in the Texan who comes with the speech synthesizer saying "one two three". We would prefer him to say, of course, "one hundred and twenty-three".

One of the clever Swedes, Anders Persson, has written a program to remedy that situation. It is written in the form of a subprogram, usually kept in merged format, to be merged with any EXTENDED BASIC program that might need it. As with all subprograms, it is legal only in a program and thus cannot be accessed directly from the keyboard.

The correct format to use it is CALL SAY_NUM(#) in which # may be any NUMERIC (as opposed to ALPHANUMERIC) data between 0 and 999. For example, to test it out, you could write:

```
100 CALL CLEAR
110 CALL SAY("999")
120 CALL SAY_NUM(999)
130 END
```

If you merge NUMTALK into this short program and RUN it, you will notice the difference between the two statements right away. As a consequence of this subprogram being able to use numerical data, it is also possible to use a variable for #, above. For example, numbers can be spoken in sequence, something CALL SAY cannot do at all. The following demonstrates this:

```
100 CALL CLEAR
110 FOR I=499 TO 500
120 CALL SAY_NUM(I)
130 NEXT I
140 END
```

I can foresee a space program with sequences such as:

```
100 FOR I=5 TO 0 STEP -1 :: CALL SAY_NUM(I) :: NEXT I
110 CALL SAY("TAKE+OFF")
120 END
```

---

To save NUMTALK in merge format: SAVE "DSK1.NUMTALK",MERGE

To merge NUMTALK into your program:

(1) load your program first

(2) enter command, MERGE DSK1.NUMTALK

```
25000 !NUMTALK, a subprogram which allows to pronounce numbers correctly in a CA
LL SAY statement
25010 ! Can be used in a program only. Correct format to be used is : CALL SAY_N
UM(#).
25020 ! # Can be any NUMERICAL data between 0 and 999, which CALL SAY will not a
ccept anyway.
25030 ! Keep NUMTALK in merged format, to be merged with any program that might
need it.
25040 ! Anders Persson, Lund, Sweden
25050 SUB SAY_NUM(NR)
25060 IF INITED THEN 25120
25070 DIM TEXT$(33)
25080 RESTORE 25370
25090 FOR I=1 TO 33 :: READ TEXT$(I):: NEXT I
25100 NUMPOS$="-+.E0123456789"
25110 INITED=-1
25120 NUM$=STR$(NR)
25130 IF ABS(NR)>=1000 OR ABS(NR)<10 THEN 25210
25140 NEG=(NR<0)
25150 IF NEG THEN NUM$=SEG$(NUM$,2,LEN(NUM$)):: NR=ABS(NR):: CALL SAY(TEXT$(1))
25160 IF NR>=100 THEN GOSUB 25240 !SAY HUNDREDS
25170 ON ERROR 25400
25180 IF VAL(NUM$)>=20 THEN 25300 !SAY TY'S
25190 IF VAL(NUM$)>=10 THEN 25350 !SAY TEENS
25200 !SAY DIGITS
25210 FOR I=1 TO LEN(NUM$):: CALL SAY(TEXT$(POS(NUMPOS$,SEG$(NUM$,I,1),1))):: NE
XT I
25220 SUBEXIT
25230 !SAY HUNDREDS
25240 SPEAK$=TEXT$(POS(NUMPOS$,SEG$(NUM$,1,1),1))&TEXT$(33)
25250 IF SEG$(NUM$,2,2)<>"00" THEN SPEAK$=SPEAK$&"+AND"
25260 NUM$=STR$(VAL(SEG$(NUM$,2,LEN(NUM$)))):: IF NUM$="0" THEN NUM$=""
25270 CALL SAY(SPEAK$)
25280 RETURN
25290 !SAY TY'S
25300 SPEAK$=TEXT$(VAL(SEG$(NUM$,1,1))+23)
25310 IF SEG$(NUM$,2,1)<>"0" THEN SPEAK$=SPEAK$&"+"&TEXT$(POS(NUMPOS$,SEG$(NUM$,
2,1),1))
25320 CALL SAY(SPEAK$):: NUM$=SEG$(NUM$,3,LEN(NUM$))
25330 GOTO 25210 !TO SAY DIGITS
25340 !SAY TEENS
25350 CALL SAY(TEXT$(INT(VAL(NUM$))+5)):: NUM$=SEG$(NUM$,3,LEN(NUM$))
25360 GOTO 25210 !TO SAY DIGITS
25370 DATA NEGATIVE,,POINT,E,ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE
25380 DATA TEN,ELEVEN,TWELVE,THIRTEEN,FOURTEEN,FIFTEEN,SIX+TEEN,SEVEN+TEEN,EIGHT
+TEEN,NINE+TEEN
25390 DATA TWENTY,THIRTY,FORTY,FIFTY,SIXTY,SEVENTY,EIGHTY,NINETY,+HUNDRED
25400 RETURN 25410
25410 ON ERROR STOP :: SUBEND
```

# TI TIPS

1. EXTENDED BASIC WITH 32K - To speed up program

```
100 CALL INIT :: CALL LOAD(-31878,0)
```

Zero in call load statement means you are not using any sprites. If you replace the zero with a seven, it allows the use of sprites #1 thru #7. By using less than the maximum of 28 sprites, you significantly increase the speed of your program.

2. To improve the sound of musical tones(from TIGERCUB)

```
100 CALL SOUND(D,N,V,N*1.01,V)
```

3. To print a slashed zero on the screen(from TIGERCUB).

```
1 CALL CHAR(48,"003A444E546444B8")
```

4. In EXTENDED BASIC, blank spaces are not required before and after the :: in multiple statement lines.

```
100 CALL CLEAR::X=10::FOR I=1 TO X::NEXT I
```

5. EXTENDED BASIC WITH 32K - CALL LOAD statement to replace RUN "DSK1.LOAD". (from Siouxland 99'ers)

```
100 CALL INIT::CALL LOAD(-31961,149)
```

6. Columnizes positive and negative numbers.(from John "Jeb" Hamilton of the Central Iowa Users Group)

```
100 PRINT TAB(C-POS(STR$(X)&".",".",1)-(X<0));X
```

C=column for decimal
X=number to columnize

7. CALL KEY that reads as upper case even if alpha lock is up.

```
100 CALL KEY(3,K,ST)
```

8. To find the number of days in any month of any year.

```
100 D=VAL(SEG$("312831303130313130313031",M*2-1,2))+(ABS(M=2)*ABS(
Y/4=INT(Y/4)))
```

D=number of days
M=month (number 1 to 12)
Y=year

9. For a surprise, push FCTN,space bar, H & N at the same time.

10. EXTENDED BASIC - Tired of blinking black cursor? (from TIGERCUB)

```
1 CALL COLOR(0,11,1)     - works only in programs
```

# TIPS FROM THE TIGERCUB

#21

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

I thought that my 28-Column Converter, as published in Tips #18, was finally foolproof, but someone found a way to print a program incorrectly with it!

I'm sure you know that characters 127-143, and on up to 159 in Basic, can be redefined and used in graphics. You probably also know that these redefined characters can be put into PRINT or DISPLAY AT statements, by holding down the CTRL key as you type them. If you load a program containing such redefined characters and LIST it, they will appear as blanks. If you RUN the program, so that they are redefined by the CALL CHAR statements, and then LIST it again, they will show up in their redefined form - but if you print out the program on your printer, they will still appear as blanks. So, before you publish a program, it's a good idea to RUN it and LIST it, and look for any of those gremlins.

If you do want to publish such a program, this fix will take care of it by underlining all characters that must be typed with CTRL down (except that lower case v is typed with FCTN down). It's slow, so only use it when you need to.

```
190 IF W$="E" THEN 195 :: PR
INT #2:".TL 126:94;" :: PRIN
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;" :: PRINT #2:"
.TL 92:46;" :: PRINT #2:".NF
"
195 PRINT "Does the program
contain":"redefined characte
rs above":"ASCII 126? (Y/N)"
196 ACCEPT AT(24,1)VALIDATE(
"YN"):Q$
282 IF Q$="N" THEN 290
283 FOR J=1 TO LEN(L$)
284 A=ASC(SEG$(L$,J,1)):: IF
 A<127 THEN L2$=L2$&CHR$(A):
: GOTO 288
285 IF A=127 THEN A=118 ELSE
 IF A=128 THEN A=44 ELSE IF
A=155 THEN A=46 ELSE IF A=15
6 THEN A=59 ELSE IF A=157 TH
EN A=61 ELSE IF A=158 THEN A
=56 ELSE IF A=159 THEN A=57
ELSE A=A-64
286 L2$=L2$&CHR$(27)&CHR$(45
)&CHR$(1)&CHR$(A)&CHR$(27)&C
HR$(45)&CHR$(0)
288 NEXT J :: L$=L2$ :: L2$=
""
```

That should do it, unless the number of added control characters stretches the line beyond 80 characters. Such is the case with the following, which I had to type in manually (it also contains low ASCII characters which the printer misinterprets as controls).

## TIGERCUB CHALLENGE

```
100!The Unprintable Unkeyabl
e Program!
110!To shuffle the numbers 1
 to 255 into a random sequen
ce without duplication
120!The strings contain the
ASCII characters 1 to 127 an
d 128 to 255
130!Most of the ASCII charac
ters below 32 or above 159 c
annot be input from the keyb
oard
140!So how was this program
programmed?
150 M$="
              !""#$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKL
MNOPQRSTUVWXYZ[\]^_`abcdefgh
ijklmnopqrstuvwxyz{|}"
160 M2$="




170 M$=M$&M2$
180 L=LEN(M$):: RANDOMIZE ::
 X=INT(L*RND+1):: N=ASC(SEG$
(M$,X,1)):: M$=SEG$(M$,1,X-1
)&SEG$(M$,X+1,LEN(M$))
190 PRINT N;:: IF LEN(M$)=0
THEN STOP ELSE 130
```

## GROCERY SHOPPING LIST

Are you desperate for some way to convince your wife that your computer and PEB and printer and all are not just a too-expensive plaything? Maybe this will do the job.

The first thing to do is to prepare a file of the grocery items she might want to buy. It will be especially useful if you can list the items in the sequence in which she will come to them in the aisles of her favorite store. This little program will set up the file. Type END when you are finished.

```
100 OPEN #1:"DSK1.BUYLIST",O
UTPUT
110 INPUT A$
120 IF A$="END" THEN 150
130 PRINT #1:A$
140 GOTO 110
150 CLOSE #1
```

If you have TI-Writer, you can also use that to create the file, edit it and add to it - but BE SURE to delete all the carriage return symbols and any blank lines at the end. Save it under the filename BUYLIST.

Next, this program will hopefully get your wife to actually sit down at the keyboard and try out your computer. It will go through the list and ask if she wants to buy. If she types in any quantity other than 0, it will output the item name and quantity to the printer. At the end, she will be given the opportunity to add any other items.

```
100 CALL CLEAR
110 OPEN #1:"DSK1.BUYLIST",I
NPUT
120 OPEN #2:"PIO"
130 LINPUT #1:A$
140 IF EOF(1)THEN 210
```

```
150 DISPLAY AT(12,1):A$
160 DISPLAY AT(12,LEN(A$)+2)
:"Q"
170 ACCEPT AT(12,LEN(A$)+2)S
IZE(-4):Q
180 IF Q=0 THEN 130
190 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
200 GOTO 130
210 DISPLAY AT(12,1):"ADDITI
ONAL? Y"
220 ACCEPT AT(12,13)VALIDATE
("YN")SIZE(-1):Q$
230 IF Q$="N" THEN 300
240 DISPLAY AT(12,1):"ITEM?"
250 ACCEPT AT(12,7):A$
260 DISPLAY AT(14,1):"QUANTI
TY?"
270 ACCEPT AT(14,11):Q
280 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
290 GOTO 210
300 CLOSE #1
310 CLOSE #2
320 END
```

The list will be in enlarged print, so that no one in the store will see her putting on her reading spectacles. And after each item and quantity is a blank square to be checked off when she picks up the item.

You might also point out that she could use the checkoff blocks to mark the items she has coupons for, and she could jot down prices on it to be sure she isn't cheated at the checkout counter, or to shop for better bargains elsewhere.

The program is set up for the Gemini printer. You may need to change the "PIO" to the name of your printer, and other printers may not have the open block character CHR$(175) available.

Of course, you can also use this program for more important things, such as shopping for computer software....!

If you type the period key while holding down the CTRL key, the printer interprets the resulting blank space as CHR$(27), even though the computer knows it is really CHR$(155). Since CHR$(27) is the ESC or "escape code" which tells the printer to interpret the following characters as function command codes, you can for instance set up the printer for emphasized double-struck double-width underlined italics by OPEN #1:"PIO" :: PRINT #1:" E G W"&CHR$(1)&" -"&CHR$(1)&" 4 ", using CTRL . in the blanks. I have been overlooking another very useful feature, the skip-over perforation. PRINT #1:" N"&CHR$(6), again with CTRL . in the blank, causes the paper to advance to the top of the next page when there are only 6 lines left at the bottom of the page (providing that you started at the top, of course). This makes it possible to LIST "PIO" a program, or PF PIO from TI-Writer Editor, without printing right across the perforations.

Ghosts! Did you ever read data from a file, and find that you were getting data from a file that was no longer on the disk? It can happen, at least if you are reading from a RELATIVE file in the UPDATE mode. When you delete a file, only its address is actually deleted - the data remains on the disk until it is overwritten by a new file. If the new file is shorter than the old one, and you try to read beyond the end of the file, you may awaken the ghost!

Are you making use of those special characters that are available on your Gemini printer? You didn't know about them? Try this. 100 OPEN #1:"PIO" :: 110 PRINT #1:" (hold down the CTRL key and type 1234567/ and then hold down the FCTN key and type <>/0;BHJKLMNQY ) ". RUN . Surprised? Some of those can be very useful, such as the true division sign that you get with FCTN H. There are many more of these that you can access by CHR$. For a complete list of them and their CHR$ codes, run this - 100 OPEN #1:"PIO" :: FOR CH=160 TO 254 :: PRINT #1:CH;CHR$(CH);:: NEXT CH :: CLOSE #1. Unfortunately, these can't be used out of TI-Writer.

Here's a handy little routine to practice up on your typing.

```
100 CALL CLEAR
110 CALL CHAR(94,"3C4299A1A1
99423C")
120 CALL SCREEN(5)
130 CALL VCHAR(1,31,1,96)
140 CALL COLOR(1,8,16)
150 FOR SET=2 TO 12
160 CALL COLOR(SET,2,16)
170 NEXT SET
180 PRINT TAB(10);"TIGERCUB"
: :TAB(8);"TOUCH-TYPING": :T
AB(11);"TUTOR": :TAB(9);" T
igercub Software": :
190 REM by Jim Peterson
200 PRINT " Watch the scree
n, not the":" keyboard!": :"
  Letters and numbers will"
210 PRINT " appear on the sc
reen grid":" in position cor
responding":" to their keybo
ard position.": :" Type the
m and they will"
220 PRINT " disappear.": : :
" Press any key"
230 CALL KEY(0,K,ST)
240 IF ST=0 THEN 230
250 CALL CLEAR
260 CALL CHAR(32,"FF80808080
80808")
270 CALL VCHAR(1,30,1,192)
280 CALL HCHAR(14,1,1,384)
290 CALL VCHAR(1,4,1,14):: C
ALL VCHAR(5,6,1,11):: CALL V
CHAR(8,7,1,6):: CALL VCHAR(1
1,8,1,3):: CALL VCHAR(8,29,1
,6)
300 CALL VCHAR(11,28,1,3)
310 CALL CHAR(48,"003A444C54
6444B8")
320 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
330 RANDOMIZE
340 K=ASC(SEG$(KEY$,INT(42*R
ND+1),1))
350 GOSUB 370
360 GOTO 420
370 X=POS(KEY$,CHR$(K),1)
380 Y=ABS(X>11)+ABS(X>22)+AB
S(X>33)+1
390 R=Y*3
400 C=((X-ABS(Y>1)*(Y-1)*11)
*2)+4+Y
410 RETURN
420 CALL HCHAR(R,C,K)
430 CALL KEY(3,K,ST)
440 IF ST=0 THEN 430
450 GOSUB 370
460 CALL GCHAR(R,C,G)
470 IF G<>32 THEN 500
480 CALL SOUND(-100,110,0,-4
,0)
490 GOTO 340
500 CALL HCHAR(R,C,32)
510 CALL SOUND(-100,1000,0,1
005,0)
520 GOTO 340
```

Here's one for the kids to have fun with. I'm sorry I lost track of who published it.

```
100 CALL INIT :: FOR J=1 TO
100 :: PRINT J :: FOR P=1000
 TO 1 STEP -J :: CALL LOAD(-
31456,P):: NEXT P :: NEXT J
```

MEMORY FULL,

Jim Peterson

MID ILLINOIS COMPUTER RESOURCE ORGANIZATION
P.O. BOX 766
Bloomington, IL 61701-0766

EDMONTON 99'ERS USER SOCIETY
P.O. BOX 11983, EDMONTON
ALBERTA, CANADA T5J-3L1

```
***********************************************************************
*        MMM   MMM   IIIIII    CCCCCCC   RRRRRRRR     OOOOOOOO        *
*        MM M M MM     II      CC        RR     RR    OO    OO        *
*        MM M M MM     II      CC        RRRRRRRR     OO    OO        *
*        MM  M  MM     II      CC        RR     RR    OO    OO        *
*        MM     MM     II      CC        RR      RR   OO    OO        *
*        MM     MM   IIIIII    CCCCCCC   RR       RR  OOOOOOOO        *
*                                                                     *
*                                                                     *
*        The MID ILLINOIS COMPUTER RESOURCE ORGANIZATION              *
***********************************************************************
```