
NEW PROGRAMS:

"POPCORN" - Tandy - MSDOS - Memory resident . 1. Dictionary 2. Calculator - Calendar - ASCII Programs. These two Programs were donated by the Radio Shack Computer Center on S. Bascom Ave.

"VISION" - MSDOS - Integrated Applications Programs. This package includes a Word Processor - Applications Manager - Graphs - Calculations, all in the Windows environment. It also requires a Hard Disk System.

"DESKMATE" - Tandy - Model IV Environment System.

"N.S.L." - Public Domain Disks (3). A set of 33 MSDOS Program Utilities that assist in filling your disks or Hard Disk with Programs that could be useful, if you need them.

"PC FOUR" - This is a Program by "Hypersoft" that allows you to operate Model IV BASIC - Machine Language Programs on your MSDOS Machines. PC FOUR works with assemblers - word processors - and other Z-80 Code Programs. The Model IV Programs are transported thru "Hypercross" and then PC FOUR runs them in the 8088 machines. The Program is a real treat for those who wish to keep their Tandy Model IV's and need to communicate in the MSDOS world. The Cost is $79.95. Ph # 919/847-4779.

TRIVIA:

There are 21 Styles of Printer Ribbons made for 58 Computer Manufacturers; to be used on 907 Models of Computers. Al LANGONE sent us this info.

PROGRAM TRANSFERRING:

A unique way to transfer Files and Programs between Disk Operating Systems without the use of Special Utilities:
TI=A, TD=A, TC=35, SPT=10, TSR=3, GPL=2, DDSL=17, DDGA=2, and FORMAT the Disk in Drive 1. The following DOS's can Read and Write to this format without changing any settings:
M4 TRSDOS, LS-DOS, DOSPLUS, L DOS, MULTIDOS, Model I NEWDOS.

SURREY MicroComputer
Users Group Newsletter

NEWSLETTER INPUT:

"The Micro Group" Newsletter is from a Model IV User's Group located in Surrey, Canada. We use inserts of their Newsletters to assist our Model IV Club Members. The editors should be thanked for producing a very informative Newsletter. We do exchange our monthly Newsletter with them. Thanks SURREY MicroComputer Users Group for assisting us throughout the year with your informative articles.

CHAOS COMPUTER CLUB Vs. N.A.S.A.

If you read the newspaper this week you may have seen that article. CHAOS was caught fooling around in N.A.S.A.'s Computer files. No, we don't condone these operations. The Club is quite notorious in Europe. We use their Services for Communications. Lucky for us we can't read German!! We do get Newsletters from their CHAOS Clubs but, I'm at a loss as to understanding them. We should not let our affiliation with them be a problem. Like all things, the bad get all the publicity and this is in turn passed on to the good also.

EDUCATIONAL PROGRAMS:

Now that the school year has started, you may wish to improve the skills of your children by checking out a few educational programs such as Spelling - Math - Word Processing, etc. They should help to improve grades. If you are continuing your own schooling, these programs can be a good review exercise.

386 WORLD:

Why is the Computer Industry rushing towards the New - 386 Computers so quickly? It appears the major reason is INCOMPATABILITY! Yes, the 32 bit bus INTEL 80386 Chip Bus demands interfacing with U.S.A. manufactured boards. Those who design their own 32 bit chip will not be compatible with the INTEL Chip Computers.

Should you be interested in the new great machines, don't buy one that does not have the latest INTEL 80386 Chip.

NOTE!!! INTEL recalled 500,000 chips due to a BIG Bug in them. Don't buy a machine that hasn't had the latest INTEL chip installed. Yes, it could ruin your whole day!

Clone manufacturers will be required to sell all 32 bit Hardwares for their computers. The Tandy 4000 has INTEL's Chips in them. It also has a reasonable price tag.

MODEL I NEWS:

Should you need to update your machines, we have the following parts available:

Expansion Interface - $150.00
New Double Density Board (Aero) - $105.00
Monitor - $25.00

Also - Bernie says he can still modify keyboards for lower case functions.

DOS Utility Tips

Major David Boyd
Command and Control Microcomputer Users' Group

Some DOS commands can get you into deep trouble if you try to use them without an intimate understanding of what they do.

RECOVER

Several files on a hard disk were recently damaged, probably by a telephone placed too close to the machine. (The bell ringer in a telephone is electromagnetically driven, so it produces a fairly powerful magnetic field when it rings. This magnetic field can damage floppies, hard drives, and — in extreme cases — can even cause problems with RAM.)

In an effort to recover the damaged files, the operator used the DOS RECOVER command. Unfortunately, that produced a far more serious problem. If RECOVER is executed against an entire file structure, rather than against a single file, it will identify every referenced sector linkage as a file and rename it FILE0001.REC (then FILE0002.REC for the next one, and so on). When it does that, it can effectively destroy the integrity of every file on the hard drive, including the copy protection markings of copy protected software, because it overwrites unmarked files with recovered files. In other words, it will destroy your installed copies of Lotus 1-2-3 and dBASE III as effectively as if you had reformatted the hard disk. You cannot do an uninstall, normally, after you have run RECOVER. You are wise to consider RECOVER a taboo DOS command!

There are, however, things you can do to reduce the likelihood of losing all your files. First, back up the hard disk regularly (at least once a week). That which you do not back up you will surely lose, and the likelihood of damage to a file is directly proportional to the amount of work you have invested in it.

CHKDSK

Periodically run CHKDSK against your hard disk to detect problems before they start. If you get back a message saying you have "F PARAMETER ERRORS," then run CHKDSK again, but do it by entering

```
CHKDSK/F
```

A minor bug in CHKDSK causes it, when if finds errors, to ask if you want the lost clusters and chains converted into files. Even if you answer yes, it won't make the conversion unless you have given the command with the /F option. Once you've done this, you will wind up with files named FILE0000.CHK, FILE0001.CHK etc., with one file for every chain. Before you delete these files, look at each one by entering this command:

```
TYPE FILE0000.CHK
```

examining each of the files. If you find one that is obviously something you want to keep (which is not very likely, because they are almost always remnants of files you have already erased), then rename it by entering

```
REN FILE0000.CHK FILE0000.KP
```

substituting the proper filenames as required. Then, once you have checked them all, delete the garbage by entering:

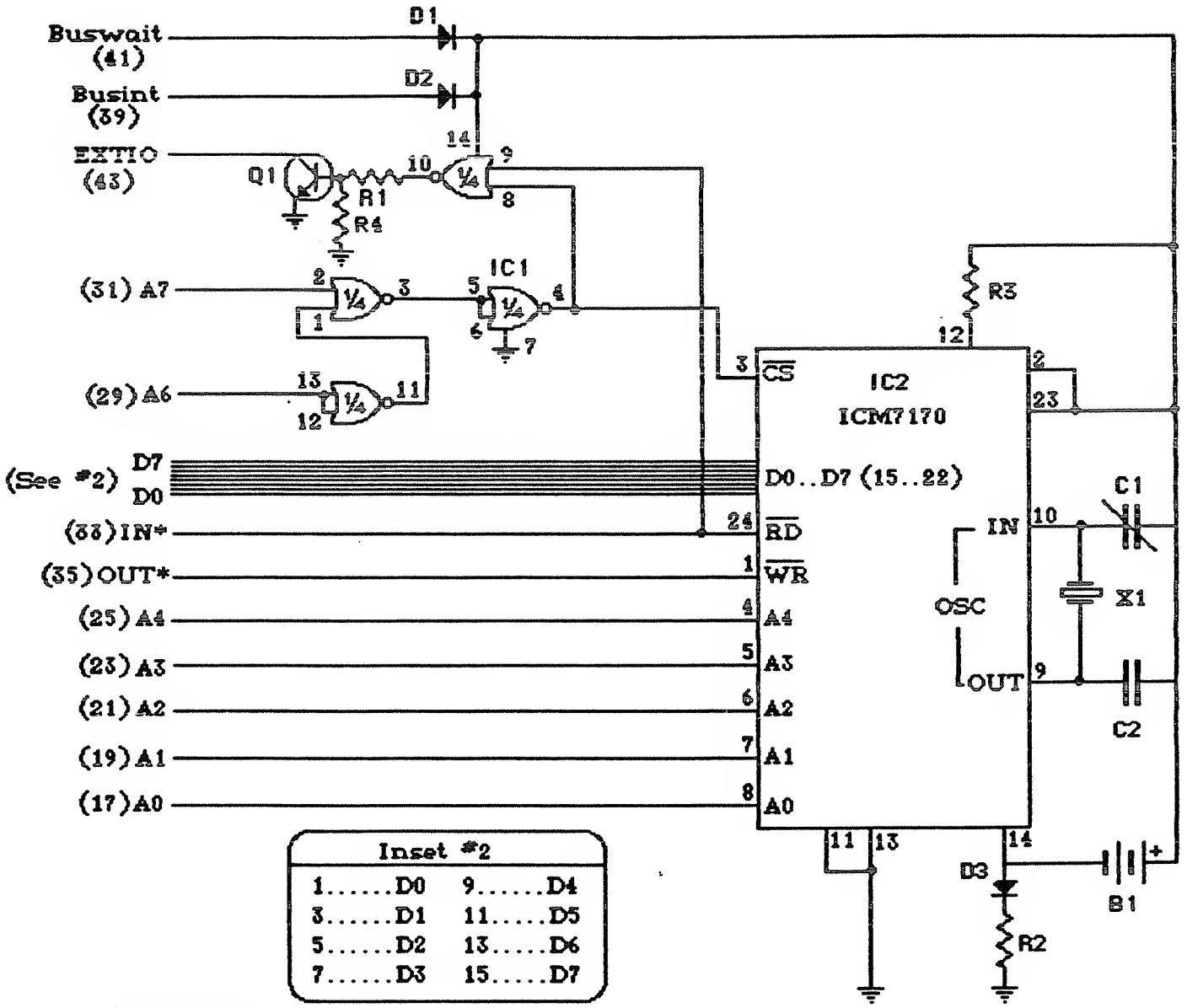
```
DEL *.CHK
```

This command will delete every file with the extension .CHK. If you ever find "F PARAMETER ERRORS" (and you frequently will after intensively editing files), you must run CHKDSK/F. These errors will grow if not treated.

BACKUP

A good safety measure, this command should be used at least once a week. It is important that you remember, however, that files transferred this way cannot be used by PC DOS or any of your application programs until they have been placed back on the hard drive with the RESTORE command.

Real Time Clock Circuit for the TRS-80 Model 4 Microcomputer



Inset #2	
1.....D0	9.....D4
3.....D1	11.....D5
5.....D2	13.....D6
7.....D3	15.....D7

Parts List

Diodes - D1,D2,D3 - IN4148

Resistors - R1 - 1K 1/4 watt
R2 - 2K2 1/4 watt, R3 - 4K7 1/4 watt
R4 - Optional (470 ohm to 1K 1/4 watt) used for stabilization only

IC's - IC1 - CD4001
IC2 - Intersil ICM7170IPG

Capacitors - C1 - 33pf trimmer
C2 - 18 pf

Crystall - X1 - 32.768 KHZ watch type

B1, 2 - 1.25V nicad batteries (see note)

Misc - Sockets, solder, wire wrap (30 awg), ribbon cable, edge card connector, etc.

Battery Note. The circuit shown uses a pair of nicad batteries in a double-A battery cell holder for backup power when your computer is off. The nicads are automatically charged whenever the computer is on. However due to the size of these batteries, you may want to use a smaller 3V lithium watch cell instead. If you do this, remove D3 and R2 from the circuit shown (they are only used to charge the nicads).

Q1 - Hard Drive Systems 2N2222 or 2N3904. Non-Hard Drive Systems 2N3393.

The following disk program was written to allow us to list programs on any printer. Your program must have been SAVEd using the ,A option (ASCII format). Simply RUN"LISTER/BAS", answer the questions, and you will be able to list programs using any line length, with neat headings and skips at perforations.

```
10 ' LISTER/BAS 2.0 — BASIC PAGE LISTER
20 ' COPYRIGHT (C) 1979 TANDY CORP.
30 CLEAR 1000 : DEFINT A-Z
32 '
34 ' ATTEMPT TO OPEN REQUESTED FILE; ERROR TRAP IF NOT FOUND
36 '
40 CLS : PRINT TAB(20) "BASIC PAGE LISTER 2.0":PRINT
50 LINEINPUT" ENTER FILESPEC: ";F$
60 ON ERROR GOTO 4000:OPEN "I",1,F$
70 ON ERROR GOTO 4100: PG=0 'PAGE COUNT SET TO ZERO
71 '
72 ' 1ST CHARACTER OF LINE IS 0 TO 9 OR 48 TO 57 DECIMAL
73 ' OR FILE IS NOT ASCIII
74 '
80 LINEINPUT#1,L$ : D= ASC(LEFT$(L$,1))
90 IF D<48 OR D>57 THEN PRINT"* NOT ASCII BASIC FILE": GOTO 50
100 LINEINPUT" ENTER TITLE: ";TL$
110 LINEINPUT" TIME AND DATE: ";DT$
120 LINEINPUT"ENTER PAGE WIDTH: ";WD$
125 IF WD$="" THEN WD=64: PRINT TAB(17); CHR$(27)WD ELSE WD=
    VAL(WD$)
130 IF WD<64 or WD>132 THEN PRINT"* BAD WIDTH (64-132 ONLY)
    *":GOTO 120
135 INPUT"DO YOU WANT EACH 'STATEMENT' ON A SEPARATE LINE";SL$
136 IF LEFT$(SL$,1)="Y" THEN F1=1 ELSE F1=0
140 PRINT:INPUT"TYPE 1 FOR SINGLE SPACE, 2 FOR DOUBLE"; SP
142 '
144 ' *** MAKE SURE THE PRINTER IS READY ***
146 '
150 PRINT:LINEINPUT"READY PRINTER — THEN PRESS ENTER > "; A$
160 IF PEEK(14312)>127 THEN PRINT"* PRINTER NOT READY *":GOTO 150
170 POKE 16424,67:POKE 16425,1 ' SETLINE/PAGE & LINE COUNT
180 GOSUB 500:GOTO 220
190 GOSUB 500
200 IF EOF (1) THEN 330
210 LINEINPUT#1,L$
215 IF F1=1 THEN GOSUB 600:GOTO200
220 W1=WD:T=0
230 W=W1:IF LEN(L$)<W THEN W=LEN(L$)
240 J=INSTR(L$,CHR$(10))' CHECK FOR LINEFEED
250 IF J>0 THEN W=J-1 ELSE IF LEN(L$)>W1 THEN J=-1
260 LPRINT TAB(T); LEFT$(L$,W)
263 IF J<>0 AND T=0 THEN T=5:W1=W1-T
265 IF SP=2 THEN LPRINT" " ' DOUBLE SPACE
270 IF J>0 THEN W=W+1
280 L$=RIGHT$(L$,LEN(L$)-W):IF L$="" THEN 310
290 IF PEEK(16425) >62 THEN LPRINT CHR$(12): GOSUB 500
300 GOTO 230
310 IF PEEK(16425) <63 THEN 200 ' PAGE FINISHED?
320 LPRINT CHR$(12): IF EOF(1) THEN 340 ELSE 190
330 LPRINT CHR$(12) ' ALL DONE
340 GOSUB 1000:GOSUB 3000:PRINT:PRINT"* END-OF-LISTING *":END
497 '
498 ' *** PRINT DASH AND HEADING ***
499 '
500 GOSUB 1000:GOSUB 2000: RETURN
600 NN=0:FOR J=1 TO LEN(L$):V$=MID$(L$,J,1):L=L+1
610 LPRINT V$;:IF L>=WD THEN LPRINT" ":GOSUB700:LPRINTTAB(10);:
    L=10
620 IF V$=CHR$(34) AND NN=1 THEN NN=0:GOTO640
630 IF V$=CHR$(34) AND NN=0 THEN NN=1
640 IF V$=":" AND NN<>1 THEN LPRINT" ":GOSUB700:LPRINTTAB(5);:L=5
650 NEXT J
660 LPRINT" ":L=0
```

(Continued on page 4)

LISTER/BAS 2.0 (Cont.)

```
670 GOSUB 700
680 RETURN
700 IF PEEK(16425)>62 THEN
    LPRINT CHR$(12): GOSUB500
710 RETURN
997 '
998 ' *** PRINT DASHED LINES FOR
    PERFORATIONS ***
999 '
1000 LPRINT STRING$(WD,"-"):
    GOSUB 3000: RETURN
1997 '
1998 ' *** PRINT HEADING AND PAGE
    NUMBER ***
1999 '
2000 LPRINT TAB(10); LEFT$ (TL$
    +STRING$(30,32),30);"    ";
    DT$;"    ";
2010 PG=PG+1: LPRINT USING "
    PAGE ###";PG : LPRINT TAB
    (10); STRING$(WD-10,"=")
2997 '
2998 ' *** PRINT TWO BLANK LINES ***
2999 '
3000 LPRINT" ":LPRINT" ":RETURN
3997 '
3998 ' *** ERROR TRAPPING ***
3999 '
4000 IF ERR/2+1=54 THEN PRINT"*
    FILE NOT FOUND*": RESUME 50
4010 IF ERR/2+1=65 THEN PRINT"*
    BAD FILE NAME *": RESUME 50
4020 PRINT:PRINT" ERROR #"; ERR/
    2+1; "IN"; ERL :STOP
4100 RESUME
```

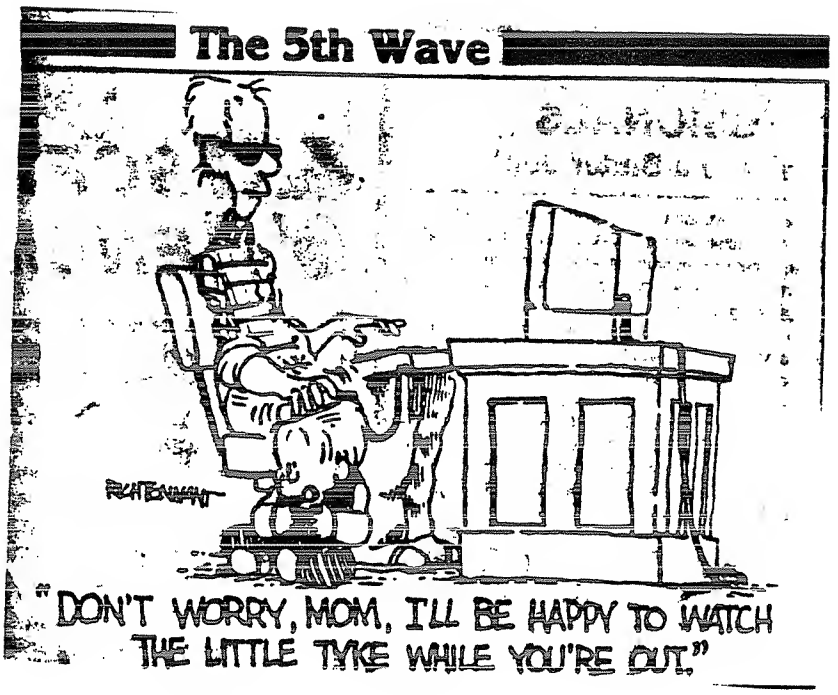

Convert Lower Case to Upper Case

If you have ever had difficulties comparing two strings where someone has accidentally used the shift key, James R. Reed of Dallas, Texas, has a solution. Mr. Reed has provided us with a short subroutine which will check each letter in a string variable to ensure that it is upper case. His subroutine is:

```
1000 A=LEN(I$):O$="":IF I$=""
     RETURN
1010 FOR B=1 TO A
1020 C=ASC(MID$(I$,B,1))
1025 IF C>95 THEN C=C-32
1030 O$=O$+CHR$(C)
1040 NEXT:RETURN
```

To use the subroutine, simply set the string you want to test equal to I$ and call the subroutine. When you return from the subroutine, set your original string equal to O$. Like this:

```
530' THE LINE BELOW WOULD BE IN
540' THE MAIN PROGRAM
550 INPUT A$
560 I$=A$:GOSUB 1000:A$=O$
570' THE REST OF THE PROGRAM
     FOLLOWS
```



Updating to LS/DOS 6.3. -Art. by K. Mohr

To add to the confusion of updating from 6.2 to 6.3, this article may help do just that!

To update SYSTEM disks, the procedure is: 1. Place 6.2 disk in drive 1, and give the command:

BACKUP :0 :1 (I,S,OLD) (enter)

(This will copy the new 6.3 system files over the old 6.2 system files)

2. If you wish to use the new BASIC files, give the command:

BACKUP BASIC/$:0 :1 (I) (enter)

(This copies Basic/cmd, Basic/ov1, & Basic/ov2 (/ov2-being new))

3. To complete update, give the command:

DATECONV :1 (enter)

(This will update the files with modification time, and when complete will give how many files where updated.)

This completes upgrading of SYSTEM disks, the next section is upgrading of DATA disks, this must be checked to verify that ONLY the BOOT/SYS and DIR/SYS exist on the disk, otherwise it is NOT a data disk!

1. Place DATA disk in drive 1, and give the command:

DATECONV :1 (CS) (enter)

(This will update the files with modification time, and when complete will give how many files where updated.)

The BASIC Language Gains Capabilities For Graphics Manipulation

by **Stewart B. Chapin,**
and **Christian E. Walker,**
True BASIC Inc.

In the early days of BASIC, when card decks and teletypewriters were the predominant data input devices, inputting and outputting numbers was a major challenge. Therefore, using graphics to display them was the least of anyone's worries. But as hardware evolved, so did BASIC. Graphics and structured programming have become important parts of what is still the world's most popular programming language.

John Kemeny and Thomas Kurtz, the Dartmouth College professors who invented BASIC (Beginner's All-Purpose Symbolic Instruction Code) in 1964, have continued to direct the evolution of the language. Recently, they created True BASIC to reflect the work done by international committees toward formulating two standards: ANS BASIC for the language and the Graphical Kernel System (GKS) for BASIC graphics. True BASIC is the first microcomputer version of BASIC to follow these new standards.

Device-independent graphics became part of the syntax of Dartmouth BASIC in 1979. Other versions of BASIC, especially HP-BASIC, also added good device-independent syntax.

True BASIC is now available for the IBM PC, Apple Macintosh, and Commodore Amiga. Version 2.0 for the IBM PC, released in the fourth quarter of 1986, puts added emphasis on graphics. For the first time, the full syntax of BASIC graphics is supported on all three IBM PC display standards: monochrome graphics, the Color Graphics Adapter (CGA), and the higher resolution Enhanced Graphics Adapter (EGA).

PLOT And WINDOW

Getting started with graphics is easy with True BASIC. For example, consider the sine function. In traditional BASIC, the function Y = sin(X), from x = 0 to x = 2π, can be examined with this short program.

```
FOR X = 0 TO 2*3.14159 STEP .1
PRINT X,SIN(X)
NEXT x
END
```

When run, this program creates a table of output for x and sin(x). However, what's really needed is the shape of the curve. In True BASIC, the graphics equivalent of the previous program would look like this:

```
SET WINDOW 0,2*pi, -1,1
FOR x = 0 TO 2*pi step .1
    plot x, sin(x);
NEXT x
END
```

The SET WINDOW statement lets the user define the coordinate scheme to be used for plotting. In this case, X is defined to range between 0 and 2π and Y is defined to range from −1 to 1. The semicolon indicates that the plotted points are to be connected in a curve; if it were omitted, only the series of dots would be shown.

In addition to points and lines, True BASIC also has syntax for ellipses, area, and text plotting. And MAT PLOT statements allow the plotting of entire arrays with a single statement.

By default, graphics (as in this example) are always scaled to fit the entire screen (or, on systems such as the Macintosh and Amiga, to fit the entire output window). But the screen can be divided into several smaller windows and different coordinate scales can be assigned to each. A more elaborate version of the sine wave example is shown in the Box.

True BASIC Example

```
OPEN #1: screen 0, .5, 0, .5      ! Open window in lower left,
SET WINDOW 0, 2*pi, -1, 1         ! assign coordinate scale
BOX LINES 0, 2*pi, -1, 1          ! and draw a box around it.

OPEN #2: screen .6, 1, 0, 1       ! Open second window—very tall
SET WINDOW 0, 2*pi, 0, 1          ! assign a different scale
BOX LINES 0, 2*pi, 0, 1           ! and draw another box.

FOR X = 0 to 2*pi step .1         ! Here's the old example
    WINDOW #1                     ! enter the first window, and
    PLOT x, sin(x);               ! draw part of the curve.
    WINDOW #2                     ! Now enter the second window,
    PLOT x, sin(x)                ! and plot unconnected points.
NEXT X

END
```

The result (see Fig) shows how advanced software can simplify graphics creation. True BASIC keeps track of window dimensions, coordinate schemes, current location, and current color. The user doesn't have to know any of these but can find them out if necessary by using queries such as ASK SCREEN, ASK WINDOW, and ASK COLOR. Thus, there's no need to pass this information between the main part of the program and graphics-oriented subroutines.

Pictures And Transformations

True BASIC also implements a new graphics-oriented unit, the Picture, which is the graphics equivalent of a subroutine. It has a name that refers to a collection of graphics statements, can take parameters, and can be called by name from other parts of the program. Pictures can be stored in external libraries, along with subroutines and functions.

But pictures have other special properties that make them different from subroutines. Besides passing parameters to them, you can apply special effects known as transformations. Built-in transformations include the ability to rotate, shift, or rescale the picture with a single statement.

In addition to the built-in transformations, you can also define your own. Pictures and transforms, together with windows and user-defined coordinates, give today's BASIC programmer extensive control, not only over what graphics output to display, but over how and where to display it.

Many people associate graphics with color. But it wasn't always so. In the early days of plotting—whether on paper or on-screen—graphics were decidedly monochrome. Paper plotters had the mechanical problem of how to change pens, but screen-oriented graphics had even greater problems. Further, color monitors are still expensive and have trouble matching the resolution of monochrome monitors.

BASIC And Color Graphics

The problem of devising the software to manage colors is a problem of at least equal difficulty. Instead of just keeping track of whether a pixel should be on or off, software now has to track what color it is and, if it's off, what color is 'off.'

Over the last five years, the trends in microcomputer graphics hardware have exemplified these difficulties. When the IBM PC was first released, it was assumed that most people would only need monochrome text displays. Later, when the Color Graphics Adapter (CGA), was introduced, its several different modes all involved tradeoffs: You can have text-only in 16 colors, medium resolution graphics in four colors, or high resolution graphics in two colors. And while the CGA user is already limited in basic color choice, there are even further limits in the choice of colors that can be used together on-screen at the same time.

The designers of the Apple Macintosh, implementing a very complex icon-based operating system, evidently decided that adding the problems of color graphics would be too difficult. Therefore, the Macintosh tradeoff involves higher resolution, but in monochrome only.

Technological advances and lower chip prices have combined to improve the graphics options of today's micro users. On newer display cards, such as the EGA for the IBM, and on machines such as the Atari ST, the Commodore Amiga, and the Apple IIGS, the hardware supports higher levels of resolution with greater color choices. Yet how does a language such as BASIC pass on those options to the user?

True BASIC programs use the SET MODE, SET COLOR, and SET COLOR MIX statements. SET MODE selects the level of resolution by using such names as 'mono' or 'high16'. SET COLOR changes the current color, which will be used by subsequent PLOT statements. Since most users don't memorize colors by number, colors can be set by name (for example, SET COLOR 'red').

On the Amiga, the EGA, and other hardware that allows it, True BASIC users can define their own colors. SET COLOR MIX (1) .1,.5,.8 sets color number one to be a mix of red, green, and blue (in this example, mostly blue-green). When the BASIC user sets the current color to 1, this blue-green shade is what will be used. SET COLOR MIX also affects the parts of an object already on-screen that have the chosen color number.

Portability

One of the important lessons learned about computer graphics over the last decade, especially on micros, is the need for portability. Applications that started on flatbed plotters might be moved to CRT displays or shared with colleagues who have output devices with different resolutions. Graphics that are hardwired to specific devices are doomed to be left behind as technology advances.

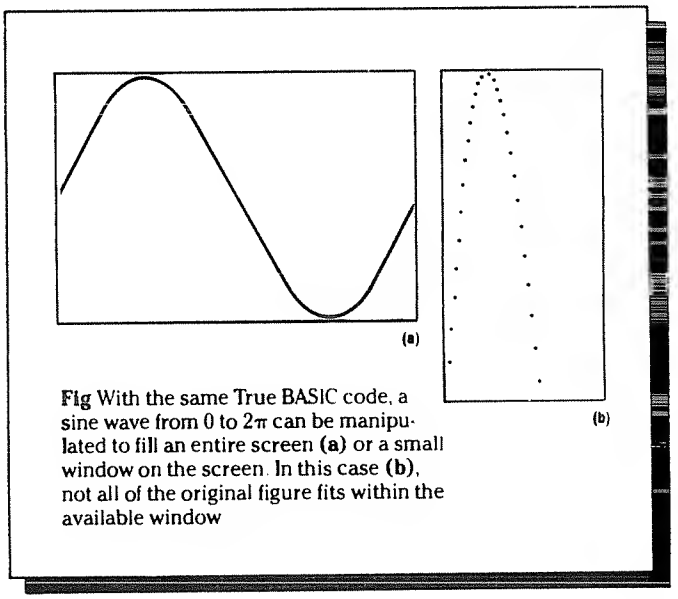


Fig With the same True BASIC code, a sine wave from 0 to 2π can be manipulated to fill an entire screen **(a)** or a small window on the screen. In this case **(b)**, not all of the original figure fits within the available window

When graphics were first introduced to BASIC at Dartmouth, the need for device-independent syntax was recognized immediately. It was a necessity because, like most schools, Dartmouth had a hodgepodge of equipment. Over the last few years, the computer world has acknowledged this problem and responded to it with standards such as GKS.

True BASIC and the ANS BASIC standard that it follows use GKS in a way that harmonizes well with the spirit of BASIC. Therefore, the ease of use that was so important when Kemeny and Kurtz first designed the language is still its hallmark, even for sophisticated graphics. ■

Simple Word Processing Program

Mr. Dave Satterfield of Carson City, Nevada sent this program for simple word processing in Level II and Disk BASIC. The program allows you to set maximum print width and equal left and right margins. While you are entering text, the number of unused characters is displayed. If you reach the right margin, a flag will come on telling you that the margin has been reached. You can type past the margin to the line length you set. At any time you can backspace one character (left arrow) or press the '@' key to erase the entire line. This program reverses the keyboard, so you shift for upper case. Remember, you can send upper/lower case to your printer if your printer supports upper/lower case, but you will not see lower case on the video unless you have the conversion kit installed. An upper case lock (shift up arrow) is provided. Press up arrow to return to upper/lower case. Pressing the ENTER key will print your text on a line printer. This program was NOT written to be used with our upper/lower case conversion kit.

```
10  CLEAR 500
100 CLS:INPUT"MAX PRINTER LINE LENGTH";L
110 CLS:INPUT"LEFT/RIGHT MARGIN";M
120 CLS:PRINT"READY":FOR N=1 TO 200:NEXT: CLS
125 C=-1:S=-1:B$="":A$="":CLS
130 C=C+1:B$=B$+A$:PRINT@128,B$;CHR$(93);:
    PRINT@40,"SPACE LEFT";L-C-M;
133 IF C>=L-(2*M)PRINT@30,"MARGIN";ELSE PRINT@
    30,"      ";
135 IF C=L-M THEN 255
140 A$=INKEY$:IF A$="" THEN 140
160 IF A$="@"THEN 125
170 IF S=1 THEN 190
180 IF(ASC(A$)<91)*(ASC(A$)>63)THEN
    A$=CHR$(ASC(A$)+32): GOTO130
190 IF(ASC(A$)<128)*(ASC(A$)>95)THEN
    A$=CHR$(ASC(A$)-32): GOTO130
200 IF ASC(A$)=27PRINT@0,
    "SHIFT";:S=1:A$="":GOTO140
210 IF ASC(A$)=91PRINT@0,"      ";:
    S=-1:A$="":GOTO140
220 IF ASC(A$)=13THEN 300
240 IF ASC(A$)=8THEN
    C=C-2:Z=LEN(B$)-1:B$=LEFT$(B$,Z):
    A$="":GOTO130
250 GOTO130
255 PRINT@0,"END OF LINE. PRINT (P) OR ERASE (E)?"
260 P$=INKEY$:IF P$="" THEN 260
265 IF P$="P"THEN 300
270 IF P$="E"THEN 125
275 GOTO260
300 LPRINTTAB(M)B$:GOTO125
```

Word Processing

TRS-80® Model I Keyboard Reverse is Now Possible with a Simple Program

Some of you who use the TRS-80 for word processing may have had difficulties using the keyboard because the upper-lower case arrangement is the opposite of a typewriter keyboard arrangement. Here is a simple program you may use to reverse upper and lower case for letters and arrows. If you have an upper-lower case printer, this program will allow you to use it more effectively.

```
10 REM ** REVERSE KEYBOARD
   (REVKEY) **
20 A$ = INKEY$ : IF A$ = "" THEN 20
   ELSE 30
30 A = ASC (A$)
40 IF A < 65 THEN A = A: GOTO 70
50 IF A < 91 THEN A = A + 32 :
   GOTO 70
60 IF A < 128 THEN A = A - 32
70 LPRINT CHR$ (A);
80 GOTO 20
```

Your video display, of course, will still be all upper case. Thanks to **Carl L. Armstrong** of Winston-Salem, N.C. for this program!

IMPORTANT NOTE FOR MODEL II USERS

After you have turned the power off to your Model II, wait AT LEAST 30 seconds before turning it back on. Failure to wait could eventually cause damage to the computer's circuitry.

Model II Diskette Caution

The TRS-80 Model II uses double-density diskettes. We have recently traced a number of serious problems to the use of lower cost single-density diskettes.

The Model II was designed to use only very high quality double-density diskettes such as our 26-4905s and 26-4906s. Extensive testing has shown that the lower cost single-density diskettes simply do not provide the quality needed to ensure reliable data transfer with the Model II.

We are not trying to say that our diskettes are the only ones that will work. There are other DOUBLE-DENSITY diskettes which work very well in the Model II. However, before you invest a lot of money, make sure that the diskettes you are buying will provide you with reliable data transfer over a period of time. Remember, when you buy Radio Shack accessories and supplies for your TRS-80, you are buying products which have been thoroughly tested and proven to work with our microcomputers.

Model II Supervisor SETBRK

The following information will allow you to "lockout" the BREAK key in Model II.

Enter DEBUG, press M to modify memory, then F000 for our starting address.

Enter the following data:

```
13 F0 21 00 00 3E 03 CF
ED 5B 00 F0 22 00 F0 EB
3E 03 CF C9
```

Check the values carefully. If they are correct, press F2, then S to return to TRSDOS. Turn DEBUG OFF. Save a copy of the routine to disk using:

```
DUMP SETBRK/CIM START=F000,
     END=F020
```

This BASIC program will demonstrate the use of SETBRK:

```
10 CLS
20 SYSTEM"LOAD SETBRK/CIM"
30 DEFUSR=&HF002
40 X=USR(0)
50 A$=INKEY$
60 IF A$="" THEN GOTO 50
70 IF ASC(A$)=1 THEN END
80 PRINT"THIS IS A TEST";
90 GOTO 50
```

This program will demonstrate that the BREAK key has been disabled. Press F1 to end the program.

KNOW YOUR D.O.S.

puters which have different capabilities. It could be said that all versions of that system are not necessarily just alike.

The Tandy 3000 family's MS-DOS 3.2 (current Ver. 3.20.02 with BIOS Ver. 1.02), for example, includes partitioning of the over-32Mb hard disks. (MS-DOS supports only 32 Mb of storage per disk.) I understand IBM didn't do partitioning until their version 3.3.

We will have a new version soon. Ver. 3.20.03 (with BIOS Ver. 1.03) will include some minor fixes, more hard disk support added to the table, and 3.5-inch 1.44Mb floppy support, for the 1.44Mb drives I told you last month were coming. MS-DOS 3.3 is scheduled around the end of this year. FYI - The Tandy 3000 and 3000 HL use the same OS, but BIOS ROMs are NOT interchangeable between the two systems.

There will be a similar new version for the 1000 SX also. Remember that although MS-DOS supplied for the Tandy 3000 and 3000 HL may run on the 1000, our release versions for the two are different. The BASIC language supplied for the 3000 does not support the 1000 family's enhanced graphics and sound capabilities.

In case you want to be sure your own system is up-to-date, the current versions of MS-DOS for the 1000 family are 2.11.25 (BIOS 1.02) for the EX, and 3.20.00 (BIOS 1.02) for the SX. The original Tandy 1000 uses MS-DOS 2.11.25 or 3.20.00, with BIOS Ver. 1.01. The Tandy 2000 current version is 2.11.03, and is not expected to change. MS-DOS for the Tandy 1200, is version 2.11.41.

WHICH SYSTEM WORKS BEST FOR YOU

If You Have This:	You Can Run in Modes:
IBM PC Convertible CGA Option	300 x 200 4-color
PC*jr* with PC*jr* Color Display	300 x 200 4-color
Portable Personal Computer and Color Display	300 x 200 4-color
Color Display and Color/Graphics Monitor Adapter	300 x 200 4-color
Color Display and Enhanced Graphics Adapter	300 x 200 4-color, 640 x 200 16-color
Enhanced Graphics Display and Enhanced Graphics Adapter	300 x 200 4-color
Enhanced Graphics Display and Enhanced Graphics Adapter (with Memory Expansion Card and Memory Kit)	640 x 200 16-color, 640 x 350 16-color
3270 PC with APA adapter	300 x 200 4-color
3270 PC/G	300 x 200 4-color
3270 PC AT	300 x 200 4-color
IBM Personal System/2 Model 50, 60 or 80 with Color Display 8512 or 8513	All modes
IBM Personal System/2 Model 30 with Color Display 8512 or 8513	300 x 200 4-color, 320 x 200 256-color
IBM Personal System/2 Display Adapter with Color Display 8512 or 8513	All modes

TH WED..MEETING..1987

JAN. DISK OPERATING SYSTEM-SYSTEMS

FEB. PROGRAMS TRANSFERING

MAR ELECTRICAL PROTECTION 110v

APL MODEM PROTICOLS

MAY UPGRADING YOUR COMPUTER

JUN COMPUTER PREVENTIVE MAINT.

JUL PRINTERS-PROBLEMS ECT.

AUG BASIC PROGRAMING

SEP COMPUTER LANGUAGES

OCT COMPUTER ACCESSORIES ECT.

NOV CLUB REVIEW 1987

DEC NEW PRODUCTS- REVIEW

THIS MEETING IS A ROUND TABLE DISCUSSION OF THE SUBJECT OF THAT MONTH.... COME AND HELP THE DISCUSSION WITH YOUR KNOWLEDGE OR YOUR PROBLEMS.....

MEETINGS:

2ND SAT.....:....1PM WORKSHOP

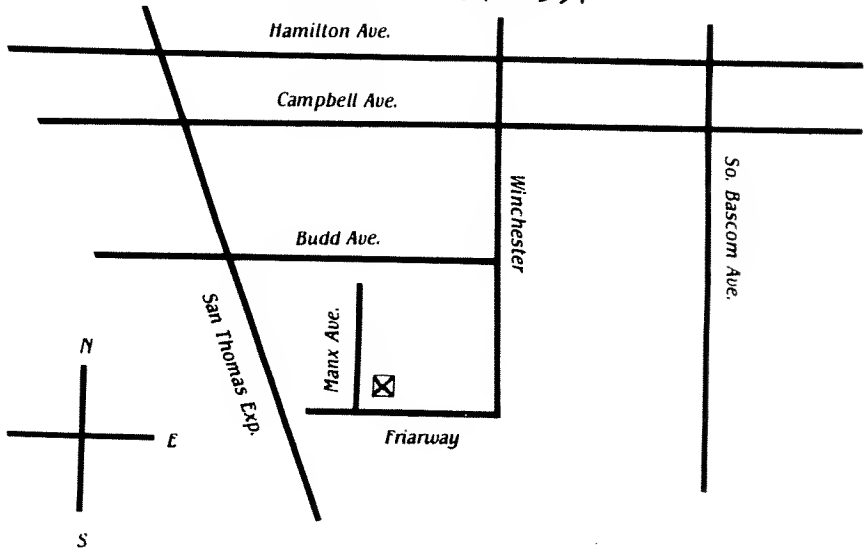
4TH WED.....7PM LECTURE

NOTES:

1. MEMBERSHIP REQUIRED
2. GUESTS..BY APPOINTMENT ONLY

INFORMATION...CALL 408-379-2774

B.B.S.....CALL 408-374-3974



This club has been established to provide its members total computer knowledge using "hands-on" application during the meetings. For better understanding of existing hardware and software in an informal—social club atmosphere.

We hope to provide members close contact between novice and amateur home computer entusiasts; young and old, male and female, school, home, business and hobbiest.

Byte Bandits of Amer.
TRS-80's Computer Clb
780 Manz Ave.
Campbell, CA 95008

SAN JOSE, CA 951
PM
1987

The Bicentennial of the Constitution of the United States of America
1787-1987
USA 22

BYTE BANDITS
CLUB
MEMBER TRS-80 MACHINE
OF AMERICA