The Boston Computer Society TI-99/ 4A User Group Meeting Newsletter October 1988

Edited by J. Peter Hoddie

The November Meeting

The November meeting of the Boston Computer Society's TI-99/4A User Group will take place on Wednesday November 16 at 7:30 PM at the Massachusetts College of Art on Huntington Avenue in Boston. There will be new disks. I have a large envelope full of material for the library. Unfortunately I had no blank disks or time to get this material to anyone to copy for the meeting. I take all the blame (as usual....). Next month could be a good one though.

The November meeting topic has been the same for 4 years running now, so why change it? The topic: that which happened at the annual Chicago TI Faire. It could be interesting again this year. Asgard has this new word processor thing (among other projects), Genial Computerware has a few tricks up its sleeve, Texaments looks to be making the big TI-Base push, MYARC is again promising a public showing of Advanced BASIC and the UCSD P-Code system, and there should be a couple really good stories that come out of this year. I can't say why the potential for good stories is particularly high this year, but trust me, it should be really entertaining.

I am still looking for copy for the newsletter. Anything is appreciated. This appeal is here for old times sake.

Thanks to those who have helped out over the past few months (in no particular order....): Justin Dowling (always doing something useful: Harrisburg last weekend!), Mike Wright, Tom Ward, Joe Rawlins, Donald Mahler, Ron Williams, and whoever I forgot. I should probably give a nod to, well maybe not.

October is a great month for concerts in Boston. Only pity is that Seiji (sp!!) Ozawa had a death in the family causing a really good concert to be replaced with Chopin. So it goes.

poT-pourri

by Mike Wright

Pseudocode is a generalized way of stating the solution to a problem that is independent of the computer. For example, you would say "increment the count value" in pseudocode. This translates to COUNT=COUNT+1 in Basic, COUNT:=COUNT+1 in Pascal, COUNT++ in C and INC @COUNT in 9900 Assembly.

— Bresenham's algorithm in pseudocode:

[The numbers in parentheses at the start of each line correspond to the Xbasic line numbers in the equivalent program. If a line number is repeated, the pseudocode is implemented in a multi-statement line.]

```
(120-150) Given a line (x1,y1) to (x2,y2)

(160) dx is the difference between the x components of endpoints
(160) dy is the difference between the y components of endpoints
(170) ix is the absolute value of dx
(170) iy is the absolute value of dy
(180) inc is larger of ix, iy

(190) plotx is x1
(190) ploty is y1 (beginning of the line)
(200) x starts at 0
(200) y starts at 0

(220) plot a pixel at plotx, ploty

(230) for all increments i from 0 through inc
(240)     increment x using ix
```

The Boston Computer Society
TI-99/4A User Group
One Center Plaza
Boston, MA 02108

```
(240)   increment y using iy
(240)   plot is false

(250)   if x is greater than inc
(250)           plot is true
(250)           decrement x using inc
(250)           increment plotx if dx is positive
(250)                   decrement plotx if dx is negative

(260)   if y is greater than inc
(260)           plot is true
(260)           decrement y using inc
(260)           increment ploty if dy is positive
(260)                   decrement ploty if dy is negative

(270)   if plot is true, plot a pixel at plotx, ploty

(280)   increment i
```

— Bresenham's algorithm in Xbasic:

```
100 TRUE=-1 :: FALSE=0
110 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
120 INPUT "X1 ":X1
130 INPUT "Y1 ":Y1
140 INPUT "X2 ":X2
150 INPUT "Y2 ":Y2
160 DX-X2-X1 :: DY-Y2-Y1
170 IX=ABS(DX):: IY=ABS(DY)
180 INC=MAX(IX,IY)
190 PLOTX=X1 :: PLOTY=Y1
200 X=1 :: Y=1
210 CALL CLEAR
220 CALL HCHAR(PLOTY,PLOTX,128)
230 FOR N=1 TO INC
240 X=X+IX :: Y=Y+IY :: PLOT=FALSE
250 IF X>INC THEN PLOT=TRUE :: X=X-INC :: IF
DX>0 THEN PLOTX-PLOTX+1 ELSE IF DX< 0 THEN
PLOTX=PLOTX-1
260 IF Y>INC THEN PLOT=TRUE :: Y=Y-INC :: IF
DY>0 THEN PLOTY=PLOTY+1 ELSE IF DY< 0 THEN
PLOTY=PLOTY-1
270 IF PLOT=TRUE THEN CALL
HCHAR(PLOTY,PLOTX,128)
280 NEXT N
290 GOTO 290
```

— Analysis of the Basic program:

Line 100 sets up two variables to reflect the values used in Xbasic for TRUE and FALSE. This is explained on p41 of the TI Extended Basic Manual.

Line 110 sets up character 128 to be a solid box. Since we cannot plot pixels in Xbasic, an 8x8 box will be used to represent a pixel.

Lines 120-150 call for the start and end line co-ordinates. x values must be between 1 and 32 and y values must be between 1 and 24. In the interests of clarity, there is no error range checking and the program will halt on error if you exceed these values.
Line 200 deviates from the pseudocode since the top left corner of the screen in Xbasic is (1,1) and not (0,0).

Line 210 clears the screen of the input statements in preparation for plotting the line.

In line 220 note that HCHAR uses (row, col, char) which corresponds to (y, x, char) and not plot at (x,y). Since CHAR 128 had all its pixels turned on in Line 110, it can be considered to be one large pixel.

Line 230 starts count at 1 instead of 0 to accommodate the Xbasic screen co-ordinates.

Lines 250 and 260 show the power of multi-statement lines. The portions following the IF...THEN test are only executed if the test is true. In the pseudocode this is shown by the indenting of the statements below "if x is greater..." and "if y is greater...".

Line 280 increments the FOR...NEXT variable which is equivalent to "increment i" in the pseudocode. Note the tendency of many programmers to use i or I as the loop variable. Unfortunately this causes problems in printed listings because it is easily confused with 1 (one) or a lower case l (ell). Similarly, you should avoid using O or o as a variable, since it too is easily confused with 0 (zero).

Line 290 freezes the program leaving the plotted line on the screen. You should press FCTN-4 to break to command mode.

This Xbasic program will allow you to input two screen co-ordinates and will plot the best line between them. It would be easy to take the program and convert it to a sub-program that would require the screen co-ordinates to be passed to it.

[This article is based on material in the book "Advanced Graphics in C: Programming and Techniques" by Nelson Johnson. Osborne McGraw Hill, copyright 1987, ISBN 0-07-81257-7. In a letter dated September 19, 1988 (ref 1205) Osborne McGraw Hill has generously granted the BCS permission to reproduce this material. Although the book deals primarily with implementing graphics on the Enhanced Graphics Adapter (EGA) used on IBM-type computers, it contains many useful ideas and is written in a clear and informative style. The book is aimed at the intermediate to advanced-level C programmer.]

Four into one does go

Multi-Mod is an upgrade kit for Triton's Super Extended Basic cartridge that gives you access to the Editor/Assembler, Disk Manager III and TI-Writer, in addition to Super Extended Basic.

This means that with only one cartridge you have access to the four most frequently used TI cartridges, or modules, hence Multi-Mod. With Multi-Mod you can probably do without a Widget, and will also probably reduce cartridge port wear and tear.

The E/A and TI-Writer in Multi-Mod are the same as those in the original TI cartridges. However, DM III will allow you to access up to DSK9 and DSDD drives. In addition DM III has help screens with complete listings of error codes and the PIO port can be selected from the menu. DM III is compatible with the TI, CorComp and Myarc disk controllers.

You only need a screwdriver to install Multi-Mod. It is used to open the cartridge. No soldering is required.

The Multi-Mod upgrade kit includes a chip, an instruction manual describing how to use Multi-Mod's features, and a diskette with support files for the Editor/Assembler and TI-Writer. Manuals for the added cartridges are not supplied, but are still available from TI.

To order, send $22.95 (cash or check) to:

John Guion,
11923 Quincy Lane,
Dallas, TX 75230.

New TI publication available

Bruce Forbes of Sumerduck, VA is producing a new bimonthly TI publication called Tid Bits. It is available in two media: hardcopy or disk.

The hardcopy version is $8 for 6 issues and the disk version is $12 for six issues (SSSD flippy). All of the material in the hardcopy version is available on disk, except for the illustrations. However, the disk version will sometimes contain extra public domain software.

Bruce's idea is to produce at least six issues while looking for a sponsor or advertising support. If successful, he plans to continue the publication.

The address to become a subscriber is:

Bruce Forbes — editor,
Route 2, Box 412,
Sumerduck, VA 22742.

One hundred and FORTI two steps

One of the last projects in the works before TI pulled out in 1983 was the FORTi music card. The card gave the 4A a 12-voice, 2 or 4 channel synthesizer capability. Using a Forth-like language, you could control all 12 voices simultaneously, with bass as low as 32Hz.

I understand that the plans for the card were used by former TI employees to manufacture about 100 units. They were last advertised in Micropendium in June 1985 at a cost of $200.

Now Mike de Frank has completed a 142-step set of instructions that will enable you to build a FORTi music card on the Coleman/Williforth Protoboard.

To get your copy of the instructions send a SASE to John F. Williforth, RD #1 Box 73A, Jeannette, PA 15644.

If you are not familiar with the Protoboard, it was announced in the December 1987 issue of the newsletter of the West Penn 99'ers. Delivery of the board started in September 1987 and at the time of the announcement about 200 had been delivered. It is available in quantities 1-4 at $35 each + S/H (PA residents add 6% sales tax) from:

The Computer Bug,
5075 Clairton Blvd,
Pittsburgh, PA 15236
(412)-882-3374

For larger quantities call:

Scott Coleman,
823 North Ave,
N. Braddock, PA 15104
(412)-271-6283

For technical information on the Protoboard you can contact John Williforth at (412)-527-6656.

3.5-in drive kit

Alpha Scientific is offering a 3.5-in drive kit for the TI-99/4A. It consists of a drive, mounting hardware and facing, and only requires a screwdriver to install. The drive is used exactly the same as your current 5.25-in drive. The 3.5-in drive is half-height, so two would fit in your PE box.

For further details, write to or call:

Alpha Scientific,
PO Box 626,
Chesterfield, MO 63006.
(314)-878-7117

Goodbye Monty

The September 1988 newsletter of the Madarea 99ers reports that Monty Schmidt has sold his TI systems and has used the money to buy a PC clone.

Monty came to national attention when he offered his Techie bulletin board system at no charge. He was responsible for Command DOS, and also wrote "Technical Drive", a book that described many of the inner workings of the DSRs in peripheral cards.

Monty was an extremely personable and easy-to-like TI-er. His expertise and skills will be sorely missed, but we wish him well.

ROM vs GROM

In the days of the 99/4 the console power-up code did not search ROM for a GROM header. This meant that every cartridge designed for use with the 99/4 had to have at least one GROM chip.

This was one of the changes TI made when designing the 99/4A. The 4A has the capability of executing ROM-only applications. The 4A power-up code checks CPU memory location >6000 (usually referred to as ROM-in-GROM) for a valid ROM/GROM header. It then searches the linked list of user programs (if any) indicated by the header. These user programs are added to the applications selection list. Most 99/4As have this capability; some do not.

The 4A code searches ROM first, and then GROM. But, to maintain compatibility with the 99/4, TI displays the selection list in reverse order — from the last found GROM application to the first found ROM application.

If you have a device such as the Gramulator, it will always appear at the bottom of the selection list — well until this week anyway.

Mark van Coppenolle, the designer of the Gramulator and its supporting software, has available a 34-sector file called RAMFIRST. When this is loaded into the Gramulator's GRAM 0, and the TI-OS/GRAM 0 switch is set to GRAM 0, it changes the power-up code so that the contents of ROM (or in the case of the Gramulator, RAM), are displayed first.

This is a nice touch for someone with a CorComp controller in that it will always allow you to select the Gramulator from the master selection list.

For more details write to: CaDD Electronics, 52 Audubon Road, Haverhill, MA 01830. Telephone 603-895-0119.

Rave 99 MX01 memory enhancement — 2

On October 2, Rave's John McDevitt took time out to drive from Vernon, CT to Andover, MA to be the guest speaker at the monthly meeting of the Magnetic user group.

The core of the material he presented was covered in the review which appeared in the June issue (p5). However, there were some additional interesting sidelights, all of which were presented in a knowledgeable, informative and entertaining way.

+ The supercap used to maintain power on the card was a spinoff from work done by a large oil company during the oil crisis. It seems they were trying to develop alternate energy sources to power cars. Instead of using batteries they turned to capacitors, and the supercap was born. Incidentally, the supercap uses a chemical process to retain its charge. Even when the voltage starts to drop, this generally reflects the state of the outside cells. The inner cells are usually at full voltage. This is why it will recharge after the power in the P-box has been on for only two or three minutes.

+ The Rave card does not currently work with Myarc's Geneve 9640 computer. Rave has written the code to make the card compatible and has offered to supply it to Myarc for inclusion in MDOS. However, Myarc's answer is that they are "running out of room" in MDOS.

+ So far Rave has sold 50 of the MX01 memory enhancement cards.

+ The ramdisk software allows you to define up to 10 disks on a single card. These disks can have odd sector sizes and can be named. (An "odd" sector size is not a multiple of 360.) For example, you can have one disk called TIMP which exactly contains the Multiplan files, one called TURBO that exactly contains the Turbo Copy files, and so on. These programs require that the disk have a specific name. When you need to use such a program, simply move that disk into slot 1.

+ There is a new release of the Rave software due very soon now. Some of the features include the ability to do the equivalent of the Rave-supplied Basic CALL statements from the auto-boot screens. For example, a CALL AO or CALL AF can be issued without having to go to Basic. In addition, there is some useful status information that will show you what disk has been assigned to what slot. Rave has also improved the BOOT program and claims that it is now unable to crash it.

+ As a user you can use the CONFIG program to reserve a memory bank for your own use. In Rave's terminology, a "bank" is 32K. Rave does this itself by reserving four user banks when running the equivalent of the Myarc XBII in the card, without the need for the Myarc cartridge.

+ The print spooler is still a planned product, and will be forthcoming when Rave manages to corner J. Peter Hoddie and coax the code from him.

+ Finally, if you have more than one Rave card in the P-box, you do not have to concern yourself with knowing about the physical card. If a large ramdisk is assigned it can physically span two cards, but from an application point of view it is treated as one disk. This is accomplished by some smart footwork in the Rave ramdisk software.

John McDevitt also told the meeting that Rave was having trouble sourcing the 99/105 keyboard. It is possible that no more of these keyboards will be made. So, if you're thinking of buying one, don't delay too long. John felt that there were sufficient units in the hands of the bigger dealers to cope with demand.

For details on how to attend Magnetic meetings or how to join the group contact Joyce Corker at 617-891-1397.

Ryte Data has a new name

According to Nick Iacovelli Jr, writing in the September issue of the Chicago Times:

"I wrote a letter to Ryte Data because I heard rumors of them declaring bankruptcy. Bruce Ryan, president of the company, said NO. They are still supporting the TI with 3 products: Command DOS, Monty Schmidt's Technical Drive, and the 99AT expansion system. All other products have been discontinued.

"Command DOS can be purchased through Triton Products at 800-227-6900; the 99AT and Technical Drive by calling 705-457-2774.

"Ryte Data is now Millennium Computers and is active in the IBM/compatible business system on a local level."

Intro to the UCSD P-System
By Ron Williams

This month I will cover a different subject pertaining to the p-system. First any one using the p-system has noticed if they exit the p-system environment and run assembly programs many times after exiting the program the p-system will boot again and this is not always wanted. Well if you use funnelweb as I do, there are ways around this rebooting. I try to keep within funnelweb as much as possible as long as a program is running the p-system will not boot but it will boot if you exit to the title screen.

To exit funnelweb and go back to the title screen without the p-system rebooting you will need to assemble the following assembly code:

```
        DEF PCODE
        REF VMBW
NO      EQU >4E4F
DAT1    TEXT 'P-CODE HALT BOOT PROGRAM'
DAT2    TEXT '(C)1987 RON WILLIAMS    '
PCODE   LI R3,NO
        MOV R3,@>38FA
        LI R0,1
        LI R1,DAT1
        LI R2,24
        BLWP @VMBW
        LI R0,33
        LI R1,DAT2
        BLWP @VMBW
        RT
        END PCODE
```

This program will move the proper ASCII codes to the address the p-code card looks at to see if it should boot or not. If it finds the word "NO" ASCII codes in hex >4E4F the system will not boot and will exit to the title screen. The opposite is also true to make the p-system boot just put hex >0000 at this address and the p-system will start instead of exiting to the title screen. You may have noticed that in the assembly code it returns you to funnelweb and not direct to the title screen well this program has another use some assembly programs run from funnelweb, if this program is run just before executing them they will exit to the title screen after they are finished executing. This way you can keep working with programs outside of the p-system with not going through a p-system reboot. This program will not keep all programs from rebooting the p-system but after a while you will know the programs it will work with and the programs it will not work with. I use this stop boot program just before exiting funnelweb or executing another assembly program from funnelweb. When ready to exit funnelweb run this program and press FCTN QUIT instead of pressing the exit key on the funnelweb menu screen if you press the exit key the p-system will boot so that is why I press FCTN QUIT. I have put this program on my user list in funnelweb and that way it is ready for use anytime I exit funnelweb, the program is very small and doesn't take much room on a disk.

To stop the p-system from booting while in extended basic before typing BYE or pressing FCTN QUIT you can type:

```
CALL INIT
CALL LOAD(14586,78,79)
```

The number 14586 is the address >38FA in decimal and 78 and 79 are the ASCII codes for "NO".

To make the p-system boot while in extended basic before typing BYE or pressing FCTN QUIT you can type:

```
CALL INIT
CALL LOAD(14586,0,0)
```

To check the proper values are loaded you may type:

```
CALL PEEK(14586,A,B)
PRINT A
PRINT B
```

With A and B being the values checked.

Addendum—TI-BASE Review
By Joe Rawlins

My review of TI-BASE was done useing version 1.01. I am now in posession of version 1.02, which fixes the MODIFY COMMAND bug of deleating the portion of a command file it can not load. A few other small bugs were fixed, however I did not notice them.

The READ directive will accept character literals, but they must be surrounded by " ". Version 2.0 will have a READSTRING directive to alleviate this. The width of a numeric variable includes a sign and decimal point. The sign is mandatory resulting in a minimum length of 2.

Version 2.0 should be released sometime in November 1988 [can you say Chicago? -ed]. It will allow disks 1-9, and does not require the volumn name to be TIBASE. It will load from a TI controller in about 55 seconds and from a Horizon RAM disk in about 6 seconds.

The version 1.02 fixes make this program more useable now, however I look foward to 6 second load times and not having to type quotes around screen input.

Random Ramblings

By J. Peter Hoddie

Ever since I found out that no matter how hard I try that my articles are ineligible for "pick of the month" I have stopped trying to appeal to the masses. With that rather obtuse and probably inflammatory comment out of the way, I can get to the rest of this.

This months article deals with the fact that it is the middle of the semester. This means mid–terms to you and me. These take up much time. I did find this nifty little assembly routine lying on my hard drive, however. I did it one night this summer. It was the start of what I had hoped would be an arcade game. I only got as far as writing the player routine though. So what this program does is allows you to move a ship like in TI–Invaders and shoot. It does demonstrate how to handle sprites in assembly. It does not use automotion though. This was intentional. The way it is coded, the motion will be the same speed on a 9640 or a /4a. Everything is timed off the 60 cycle interrupt. Also, since I move the "shot" each time, there is absolutely no chance of missed hits like in Extended BASIC.

I was going to give this routine to Barry Traver for the current issue of TRAVelER but I messed up and it got left out. So here it is. Enjoy.

```
       DEF   START
*
       REF   VDPWD,VDPRD
*
       REF   VWTR
       REF   VSBW,VMBW
       REF   VSBR,VMBR
       REF   KSCAN
*
SHIP   DATA >1818,>3C3C,>FFFF,>FFFF
SHOT   DATA >C0C0,>0000,>0000,>0000
*
SHIPTM DATA 0                  count down
timer for ship motion
SHIPSP DATA 1
*
SHOTFL DATA 0                  shot flag.
0=false, 1=one in the air...
*
SPATRB EQU   6*>80
LMAR   EQU   10
RMAR   EQU   240
*
START  LWPI >8300
*
* remove the "*" at the start of the next two
statements for 9640
*      LI    R0,>0808          make sure 9640
sprites are on...
*      BLWP @VWTR
*
       LI    R0,INT
       MOV   R0,@>83C4         install a user
interrupt routine
*
       LI    R0,>0300
       MOVB R0,@>8374          easy KB
*
       CLR   R0
       BLWP @VWTR
       LI    R0,>01E0
       BLWP @VWTR
       SWPB R0
       MOVB R0,@>83D4
       LI    R0,>0506     *>80 is attribute
list...
       BLWP @VWTR
       LI    R0,>0602     *>800 is character
list...
       BLWP @VWTR
*
       LI    R0,>030E     *>40 is color
table....
       BLWP @VWTR
       LI    R0,>07F1
       BLWP @VWTR
       SWPB R0
       MOVB R0,R1
       LI    R0,>E*>40
       LI    R2,16
COLTA1 BLWP @VSBW
       INC   R0
       DEC   R2
       JNE   COLTA1
*
       LI    R0,2*>800
       LI    R1,SHIP
       LI    R2,16             define ship and
shot
       BLWP @VMBW              define a ship
character...
       LI    R12,192/2         start in middle
of screen
       LI    R0,SPATRB+2
       LI    R1,>000F          character zero/
white
       BLWP @VSBW
       SWPB R1
       MOVB R1,@VDPWD
       SETO @SHIPTM            make sure ship
can move....
*
```

```
       LI   R0,SPATRB+4+2
       LI   R1,>0100             character one/
transparent...
       BLWP @VSBW
       SWPB R1
       MOVB R1,@VDPWD            set up shot
characteristics
       CLR  @SHOT                shot flag
starts as false....
*
USER   LIMI 2
       LIMI 0
       MOV  @SHIPTM,R0           is user allowed
to move now?
       JEQ  USER00                yes, so do it
       JLT  USER00                yes, so do it
       B    @USER99               no, so do
whatever else we do now....
*
USER00 MOV  @SHIPSP,@SHIPTM      reset ship
timer...
*
       BLWP @KSCAN
       MOVB @>8375,R3            get the key
       SRL  R3,8                  as a word
       CI   R3,>FF               no key?
       JEQ  USER50                yup, so ignore
user...
*
       CI   R3,'S'               left?
       JNE  USER10                nope
*
LEFT   CI   R12,LMAR             too far left?
       JLE  USER40                yea, so ignore
this stuff
       DECT R12
       JMP  USER30
*
USER10 CI   R3,'D'               right?
       JNE  USER20                nope
*
RIGHT  CI   R12,RMAR
       JHE  USER40
       INCT R12
       JMP  USER30
*
USER20 CI   R3,'Q'               fire?
       JNE  USER40
nope...nothing...
       MOV  @SHOT,R0             one already in
the air?
       JNE  USER40                yea, so forget
this
       MOV  R12,R1               get current
user position
       AI   R1,3                 orient 3 to the
left...
       ORI  R1,180*256           superimpose
current vertical position
       MOV  R1,@SHOT             make shot
```

```
alive..
       LI   R1,>0F00             shot is colored
white
       LI   R0,SPATRB+4+3
       BLWP @VSBW                give it some
color...
*
USER30 MOV  R12,R1
       ORI  R1,180*256           vertical
postion*256
       LI   R0,SPATRB
       BLWP @VSBW
       SWPB R1
       MOVB R1,@VDPWD            move it...
*
USER40

*      launched shot handling....
USER50 MOV  @SHOT,R1             get current
shot position...
       JEQ  USER80                none
active....
       AI   R1,-5*256            move it up...
       MOV  R1,@SHOT              update stored
position
       LI   R0,SPATRB+4
       BLWP @VSBW
       SWPB R1
       MOVB R1,@VDPWD            update position
on screen....
*
       ANDI R1,>00FF
       CI   R1,20                is this shot
dead?
       JH   USER60                no...
*
       CLR  @SHOT                disable the
shot
       LI   R0,SPATRB+4+3
       CLR  R1
       BLWP @VSBW                remove all
color...
*
USER60
*
USER80

USER99 B    @USER
*

INT    DEC  @SHIPTM
       RT
```

TRIAD

The following block of text is here to fill space that was left due to the last minute rush in getting this newsletter out. It also happens to describe a really neat new program by Wayne Stith that should be of particular interest to the "plain vanilla" users of the 99/4A in that it allows you to have an editor, disk manager, and terminal emulator all in memory at one time with a fair degree of integration and capability. (this has been a paragraph of comments by a "principle" in Genial Computerware, not an unbiased BCS review of any sort! -jph).

Genial Computerware and Wayne Stith are pleased to announce the immediate release of TRIAD. TRIAD is a terminal emulator, disk manager and 40-column text editor all rolled into a single program so that everything resides in memory at the same time. Once loaded, there is no further need for disk access to take advantage of TRIAD's many features.

Because TRIAD's list of features is so extensive, it would be nearly impossible to list them all here. What follows is a short list of the features in each segment of the system.

TERMINAL EMULATOR: 300, 1200 or 2400 baud; odd, even or no parity, full or half duplex. Upload/ download files via Xmodem with CRC error-checking as default, switching to checksum if necessary. Countdown counter displays number of records yet to be received/sent (in decimal). Non-TI files stored as DF128. Outgoing DF128 files can be sent with or without TI header. Program checks for available space on disk before accepting incoming file, to avoid a full-disk error ten minutes into a download. Upload ASCII text files (DV80) from disk or from a stored buffer area. Auto log to disk if desired, or save buffer on the spur of the moment, disk directory, purge buffer, reconfigure terminal parameters and a window feature to page backwards in the current buffer and also show available space. 13K buffer.

DISK MANAGER: You can do the usual—format a disk (with or without verification), rename a disk, copy a disk, sweep a disk, copy files, delete files, protect/unprotect files, rename files, view text files. Single-drive operation supported.

EDITOR: 40-column fixed mode, 13K buffer. FCTN/CNTL keys are consistent with TI-WRITER. Load/save files, print files, save to the TE buffer for later uploading, or purge the buffer. If you have just been in TE mode, whatever you saw on the screen there is available for your use here.

CONFIGURATION: Routines always in memory. Change terminal settings, screen colors, log and printer names, cursor speed, QUIT key on/off; decrease size of Xmodem buffer to avoid timeouts; auto-incrementing of logname if desired. All files written in append mode; save configuration to disk.

TRIAD will also load any EA5-type program as long as it resides in high memory. TRIAD has successfully loaded FAST-TERM, DM1000, Funnelweb, Archiver 3.02, Birdwell's disk utilities and even TRIAD itself!

Coming soon: a source code/ tutorial similar to Wayne's acclaimed KWIKFONT tutorial will be available.

TRIAD is available for $20. To order your copy of this powerful and convenient combination send $20 plus $1 for shipping to Genial Computerware, P.O. Box 183, Grafton, MA 01519. Credit card orders may be placed through Disk Only Software by calling 1-800-546-9272.