

YESTERDAY'S NEWS

VOLUME 6 NUMBER 6 Established 2016

JUNE 2021

30 Years Ago...

Historical Information taken from Bill Gaskills TIMELINE

JUNE 1991:


Texaments releases TI Artist Plus! Pak, a collection of five programs including Guidelines, Display Master, Artoons, Designer Labels, and the Artist Companion of your choice from Artist #2-13, bundled with TI Artist Plus!. Cost is \$49.95.

Richard Lynn Gilbertson and Gary Bowser reach an agreement to have Bowser's Oasis Pensive Abacutors (OPA) produce the cartridge version of RICH GKXB.

Bill Gaskill announces the end of the TI-BASE USER newsletter after attracting only 33 subscribers to the data base programming publication.

The Hunter Valley 99ers User Group of Kotara, New South Wales in Australia disbands on June 25th due to lack of newsletter contributions, lack of members willing to serve on committees and a lack of programs available locally for the TI. This group was one of the leaders in the Australian TI Community who produced one of the best newsletters in the world in the mid 1980's.

Burns-Koloen Communications creates the NATIONAL USED SOFTWARE and HARDWARE CLUB to assist computer owners of all brands, makes and models in selling unwanted hardware and software.



MICROPENDIUM
MAR/APR 1997 - VOL. 14, NO. 2
CHARLES GOOD

The person who copies disks from my user group's library for our out-of-area member has been trying to get me to use Rapid Copy for a couple of years. "Charlie," he said, "I purchased Rapid Copy at a computer show a few years ago See "RAPID", Page 3

INSIDE INFORMATION



TI CLASSROOM - Tigercub Tips #25	Page 1
TINY CALENDAR	Page 2
TI-KEYS	Page 3
MAD DOG	Page 5
CHICKEN	Page 6
GAUNTLET OF DEATH	Page 6

and it is the best disk copier I have seen. I use it all the time." At the 1996 Multi User Group Conference I met the program's author, Barry Boone, in person and decided to give Rapid Copy a try. I purchased a copy from Ramcharged Computers, the current owner of all Texaments products. To make a long story short, Rapid Copy is now the primary whole disk copy software on my 99/4A system, replacing RedisKit, my previous favorite disk copy program.

Rapid Copy comes on a SSSD disk with three versions all on the same disk for TI, CorComp and Myarc floppy disk controllers. You can load it as DSK1.LOAD from Extended BASIC, in which case the software will detect which kind of floppy controller you have and automatically load the correct version of Rapid Copy. You can also load the appropriate version as an Editor/Assembler Option 5 file, which is what I do off a RAMdisk in my 99/4A system.

From the main menu you can copy disks, format disks, catalog disks and run the Myarc or CorComp disk manager if you are using a Myarc or CorComp floppy disk controller. The first two are really handy features and are two of the reasons I now prefer Rapid Copy. The third option is not really significant because there are better disk managers out there, such as Funnelweb's Disk Review or DSKU. When formatting disks with Rapid Copy you can select single- or double-sided and (with a non-TI controller) single or double density. You can also select the interface. I just accept the default interface value. If you are using a Myarc floppy disk controller you can also select 16 or 18 sectors per track for double density. The disk catalog option is really handy! Without exiting Rapid Copy one can find out what files are on the target disk before these files are erased in the copy process. You can catalog any floppy drive, but not RAMdisks.

The copy process is easy and flexible. Before copying you can use the main menu to chose verification on/off. The See "RAPID", Page 3

TIGERCUB CLASSROOM

TIPS FROM THE
TIGERCUB
By Jim Peterson

NUMBER
25



If you didn't solve the Long Division Puzzle in Tips #24, try dividing 230709 by 835. As for the solution to the Tigercub Challenge, it was right on the same page! Try creating those DATA statements with the LINEWRITER routine. I don't know why it works, but it does.

I've been asked to print more information on the "program that writes a program". I don't have room for a detailed account, but here are the basics. If you tried my TOKENLIST routine in Tips #23 you already have a list of the token codes you will need.

I won't go into the way that the computer squishes a program line number into only two characters, but you can accomplish it with `DEF L$=CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256))`, where LN has been predefined as the value of the line number.

If you need to refer to a program line in a statement, as in `GOTO 500`, use `DEF R$=CHR$(201)&CHR$(INT(RN/256))&CHR$(RN-256*INT(RN/256))`, RN being the line number.

To print a statement or command, simply print its token character. For instance, the token for DATA is 147, so you would print `CHR$(147)`. Note that all the punctuation marks used in programming, such as `(` and `+`, are also represented

by token codes which are NOT the same as their Keyboard ASCII value.

To print a variable name, either numeric or string, just enclose it in quotes, "A" or "A\$".

To print a value, or an unquoted string (as in a DATA statement), or the word which follows a CALL, you must print `CHR$(200)` followed by a token giving the number of characters to follow, such as `CHR$(5)` for a 5-character word such as CLEAR, then the value in quotes. For instance, the token for CALL is 157, so `CALL CLEAR` is `CHR$(157)&CHR$(200)&CHR$(5)&"CLEAR"`.

You can simplify that by predefining `DEF U$(V$)=CHR$(200)&CHR$(LEN(V$))&V$`, and then simply print `CHR$(157)&U$("CLEAR")`.

A quoted string is handled in the same way except that it is preceded by token 199 instead of 200, so you can predefine it as `DEF Q$(V$)=CHR$(199)&CHR$(LEN(V$))&V$` - the computer will take care of the quote marks.

Each program line must end with `CHR$(0)`, and the last record you print must be `CHR$(255)&CHR$(255)`.

A MERGE format file is `D/V 163`, so open the file with `OPEN #1:"DSK1.MERGEFILE",VARIABLE 163`.

Don't print more than 163 characters in a record or the computer will blow its mind! You can print

multiple-statement XBasic lines, but be sure to use the double-colon token `CHR$(130)` as the separator, not two of the `CHR$(181)` colon tokens.

Any errors you make will usually not show up until you try to MERGE or use the program you have created. `I/O ERROR 25` means that you forgot the final `255 & 255`; `DATA ERROR` or `SYNTAX ERROR` probably means that you left off a `CHR$(0)` or gave the wrong count of characters after `CHR$(200)`.

Here's a bit of psychedelic blues - -

```
100 REM - FRANKIE & JOHNNIE
    by Jim Peterson
110 DIM S(12)
120 CALL SCREEN(2)
130 FOR R=1 TO 12
140 CALL COLOR(R+1,1,1)
150 FOR T=R TO 25-R
160 CALL HCHAR(T,R,32+R*8,34
-2*R)
170 NEXT T
180 NEXT R
190 DATA 262,294,311,330,349
,392,440,494,523,587,40000
200 FOR N=1 TO 11
210 READ S(N)
220 NEXT N
230 FOR J=1 TO 110 STEP 2
240 CALL COLOR(A+1,1,1)
250 READ T,A
260 CALL COLOR(A+1,A+2,A+2)
270 FOR TT=1 TO T
280 CALL SOUND(-999,S(A),0)
290 NEXT TT
300 NEXT J
310 RESTORE 330
320 GOTO 230
330 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,2,1,1,11,13,1
340 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,12,1
350 DATA 1,11,3,1,2,5,2,6,2,
7,2,9,1,11,1,9,2,10,4,7,1,9,
1,11,7,9
360 DATA 4,7,2,8,2,9,1,11,3,
9,1,11,1,9,4,8,2,7,6,6
370 DATA 4,4,1,11,3,4,4,3,16
,2,1,11,4,7,2,6,4,7,4,6,20,1
,8,11
```

You can too have a blank space in your disk filenames! Just use `FCTN V` for the blank, instead of the space bar. You can even have a diskfull of 10 programs with invisible filenames consisting of 1 to 10 of those `FCTN V`'s.

However, those invisible characters can do strange things when you list your disk catalog to a printer. (And they are not recognized by the Disk Manager. - Ed.)

If you want to INPUT a string with leading and/or trailing blanks, just enclose the whole works in quotation marks. Try this -

```
100 INPUT A$ !type TEST
110 PRINT A$;LEN(A$)
120 INPUT A$ !type " TEST "
130 PRINT A$;LEN(A$)
140 GOTO 100 !you can even
input a blank string of 136
characters
```

I really shouldn't tell you this, but if you want to make it difficult for someone to LIST your program, just insert a garbage line, every 5th line or so until you run out of memory, consisting of REM followed by 4 or 5 lines of random characters typed with the CTRL Key held down.

Here's a program that can actually read your mind!

```
100 CALL CLEAR
110 PRINT "TIGERCUB MIND REA
DER PROGRAM": :
120 PRINT "I'll bet you a do
llar I can guess what you ar
e thinking.": :
130 GOSUB 440
140 PRINT "And I'll bet ano
ther dollar I can tell if wh
at you are thinking is cor
rect.": :
150 GOSUB 440
160 PRINT "And I'll bet anot
her dollar I'm right BOTH ti
```

```

mes." :
170 GOSUB 440
180 PRINT "And I'll bet one
more dollar I can guess what
you'll be thinking a minute
from now." :
190 GOSUB 440
200 PRINT "OK...." :
210 GOSUB 480
220 PRINT "You're thinking t
hat a compu-ter can't possib
ly know what you are thin
king.....right?" :
230 GOSUB 480
240 PRINT "So I told you wha
t you were":"thinking.....
,right?" :
250 GOSUB 480
260 PRINT "You owe me a buck
." :
270 GOSUB 480
280 PRINT "And you're absolu
tely right..I can't re
ad your mind." :
290 GOSUB 480
300 PRINT "So I told you cor
rectly that":"what you were
thinking was":"correct.....
right?" :
310 GOSUB 480
320 PRINT "You owe me anothe
r buck." :
330 GOSUB 480
340 PRINT "So I was right BO
TH times...right?" :
350 GOSUB 480
360 PRINT "That makes three
bucks you owe me." :
370 GOSUB 480
380 PRINT "And now it's a mi
nute later":"and you're thin
king you've":"been played fo
r a sucker....":"...right?" :
390 GOSUB 480
400 PRINT "...so you owe me
four bucks." :
410 GOSUB 480
420 PRINT "NEVER NEVER bet a
gainst a computer!!"
430 END
440 PRINT "Want to bet? Type
Y(yes)" :
450 CALL KEY(3,K,ST)
460 IF (ST=0)+(K<>89)THEN 45
0
470 RETURN
480 FOR D=1 TO 800

```

```

490 NEXT D
500 RETURN

Since the manual
doesn't mention it, some
folks don't know that you
can use IMAGE and PRINT
USING for output to the
printer. Try this -
100 OPEN #1:"PIO"
110 INPUT "NAME?":N$
120 INPUT "AMOUNT?":A
130 PRINT #1,USING "#####
#####"
.#":N$,A
GOTO 110

```

Of course, you could also add a line -

```

105 IMAGE "#####
#####.##"

```

And change line 130 to

```

130 PRINT #1,USING 105:N$,A

```

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a returnable mailer WITH RETURN POSTAGE, at least - and I hope you'll also include something more!

Attention, assembly programmers! Fred Hawkins of the Lehigh UG is trying to coordinate a project of documenting the operating system by breaking the console ROM down to pages of 256 bytes so that each individual or group can work on just one page. Only those who participate will

share in the results! All this is far beyond me, but if you want in, send an SASE and a SSSD disk with return postage and mailer to Fred Hawkins, 1020 N 6th St, Allentown PA 18102 - soon!

If you have a program on disk which is so long that you must type CALL FILES(1) before you can load it, add several program lines to it consisting of REM and any key you want to hold down for 5 lines. Then SAVE it back to the disk; it will now be in INT/VAR 254 format and will load without CALL FILES(1). If you then need sometime to make a cassette copy, just delete those lines and SAVE it back to disk again.

If a program loads, but gives you a MEMORY FULL IN LINE ... when you try to run it, it has used up all available memory while reading DATA into arrays or performing other internal calculations. If it runs for some time and then gives you the MEMORY FULL message, it is because you have repeatedly jumped out of a FOR...NEXT loop with an IF...THEN...GOTO before the loop is completed. This rarely happens but it can, especially when you repeatedly jump out of the innermost of several nested loops.

MEMORY FULL-Jim Peterson

TINY CALENDAR

```

1 DIM T(12),D(12),MO$(12)::
CALL CLEAR :: CALL SCREEN(2)
:: FOR I=0 TO 14 :: CALL COL
OR(I,16,2):: NEXT I
2 FOR I=1 TO 12 :: READ T(I)
,D(I),MO$(I):: NEXT I
3 DATA 7,31,JANUARY,30,28,FE
BRUARY,8,31,MARCH,32,30,APRI
L,9,31,MAY,32,30,JUNE

```

```

4 DATA 9,31,JULY,31,31,AUG,6
,30,SEPTEMBER,30,31,OCTOBER,
7,30,NOVEMBER,30,31,DECEMBER
5 DISPLAY AT(5,14):"TINY":"
EPSON/T.I. CALENDAR":"":
":"**THIS PROGRAM WILL PRIN
T A":" CALENDAR FOR ANY YEA
R FROM":" 1776 TO 2099."
6 DISPLAY AT(13,1):"**SET TO
P OF FORM AND ENTER":" THE
YEAR AS A FOUR DIGIT":" NUM
BER (ex. 1985) OR":" JU
ST ENTER TO EXIT PROGRAM"
7 DISPLAY AT(19,1)BEEP:**EN
TER CALENDAR YEAR" :: ACCEPT
AT(19,24)SIZE(4)VALIDATE(DI
GIT):Y$
8 IF Y$="" THEN CALL CLEAR :
: END ELSE Y=VAL(Y$):: IF Y<
1776 OR Y>2099 THEN 7
9 IF INT(Y/4)*4=Y AND NOT(IN
T(Y/100)*100=Y AND INT(Y/400
)*400<Y)THEN D(2)=29
10 DI=Y-1906+INT((Y-1901)/4)
:: D(0)=DI+1-(INT(DI/7)*7)
11 M2=0 :: OPEN #1:"PIO" ::
PRINT #1:CHR$(27);"S";CHR$(1
);CHR$(15);CHR$(27);"3";CHR$
(14):TAB(19);Y
12 FOR I=1 TO 12 STEP 2 :: P
RINT #1:TAB(T(I));MO$(I);TAB
(T(I+1));MO$(I+1)
13 J,K=1 :: A,M1=D(I-1)+M2 :
: B,M2=M1+D(I)
14 PRINT #1:CHR$(27);"3";CHR
$(8);"S M T W T F S
S M T W T F S":CHR$(2
7);"3";CHR$(14);"- - - -
- - - - - - - - - -
-"
15 IF J>D(I)THEN 16 :: IF A>
7 THEN A=A-7 :: GOTO 15 ELSE
PRINT #1:TAB(A*3-2);STR$(J)
;:: IF A=7 THEN 16 ELSE A=A+
1 :: J=J+1 :: GOTO 15
16 IF K>D(I+1)THEN 17 :: IF
B>7 THEN B=B-7 :: GOTO 16 EL
SE PRINT #1:TAB(21+B*3);STR$
(K);:: IF B=7 THEN 17 ELSE B
=B+1 :: K=K+1 :: GOTO 16
17 IF J>D(I)AND K>D(I+1)THEN
18 ELSE PRINT #1:"" :: A=A+
1 :: B=B+1 :: J=J+1 :: K=K+1
:: GOTO 15
18 PRINT #1:"" :: NEXT I ::
PRINT #1:"":CHR$(27);"e" ::
CLOSE #1 :: RESTORE :: GOTO
2

```

default is off. You are prompted for the drive numbers (1-4) that contain the master and target disks. You can use the same drive number for both master and target disks if necessary and are then prompted to insert each disk as needed. If you have a Myarc RAMdisk Rapid Copy will recognize it as drive 5. Other types of RAMdisks are not supported, and, no, you can't copy to multiple drives. Only one target disk at a time is allowed. Too bad. If you are using a TI controller you are also prompted whether or not you want to format the target disk. With other types of controllers the target disk is always automatically reformatted whether or not it has previously been formatted. Press the "any Key" and the copy process begins. First the target disk is formatted, then 5 (DD disks) or 10 (SD disks) tracks are read into memory and then written to the target disk. An on-screen display shows the current track being formatted/read/written. I did some timed tests to check on copy speed. Using a CorComp controller and copying an almost full DSSD disk, RedisKit (which like Rapid Copy always formats the target disk) took 1 minute 18 seconds. Rapid Copy with verification off took 1 minute 24 seconds. Rapid Copy with verification on took 1 minute 39 seconds. With a TI controller these times will be slightly longer.

Verification is an important option in Rapid Copy and one of the reasons I now prefer Rapid Copy to RedisKit. RedisKit has some error trapping and will sometimes abort when it finds a disk error, which is good. But rarely RedisKit fails to detect a disk error and makes a bad copy without telling you. So far this has never happened to me with Rapid Copy if I have verification "on." If Rapid Copy detects an error it doesn't just abort the copy operation, it reports an error and asks you if you want to continue anyway and ignore the error. Sometimes ignoring is a good idea, particularly if the error is in the master disk.

The two features I like most about Rapid Copy are its ability to catalog target and master disks from within the program and its ability to verify and make sure you have a good copy. Rapid Copy isn't quite as fast as RedisKit, but the above two features make the small amount of additional copy time worthwhile to me. Rapid Copy is commercial. You can get it with a hard copy user guide for \$9.95 from Ramcharged Computers.

TI-KEYS

BB&P - Sep 1988 - Charles Good

A "hot-Keys" program, once loaded, allows you to execute commands with just a single keypress. For example, you might press CTRL/R and have RUN "DSK1. appear on the screen waiting for you to type the rest of the file name and then <ENTER>. Such one keypress (hot key) text strings are called macros. For the TI there are three hot key programs that I know about. EZ-KEYS is a commercial offering from Asgard Software. I haven't had much opportunity to use EZ-Keys, but it has received excellent

reviews in Micropendium (January 1988) and in the newsletters. Apparently EZ KEYS is much more than just a "hot Keys" program and can best be described as an XBASIC programming environment. I have had the opportunity to play with the commercial program SOFT-KEYS from Quality 99 software. SOFT-KEYS only allows you to configure the number Keys as hot Keys via CTRL/n. Because of the limited number of configurable Keys I consider SOFT-KEYS to be inferior to both EZ-KEYS and the third "hot Keys" program TI-KEYS.

TI-KEYS is fairware and can be found in most user group libraries. The requested fee of \$10 is less than the cost of EZ-KEYS and about the same as SOFT-KEYS. Unfortunately I haven't been able to pay the TI-KEYS author Wes Johnston. He has moved from the address listed in the docs (404 Furman Lane, Ladson NC) and my check was returned by the post office with the notation "Forwarding time expired". Wes, if you read this, write me in care of the Lima UG and let me know your current address.

When booted from a ramdisk and used in conjunction with FUNNELWEB, TI-KEYS adds some really useful capabilities to the 99/4A.

You boot TI-KEYS as LOAD or as a CALL LOAD from XBASIC. A set of predefined macros is loaded which is accessed by pressing CTRL and another Key. These predefined macros are show below, and are accessible from XBASIC command mode or within an XBASIC program.

- | | |
|-----------------|--------------------|
| A - ACCEPT | S - SAVE "DSK |
| B - BEEP | T - TABC |
| C - CALL | U - U |
| D - DELETE "DSK | V - VCHARC |
| E - END | W - CALL INIT |
| F - FOR | X - CALL LOADC"DSK |
| G - GOSUB | Y - CALL LOAD(- |
| H - HCHARC | Z - CALL LINKC" |
| I - IF | 1 - RUN |
| J - JOYSTC | 2 - \ |
| K - KEYC | 3 - : |
| L - LINPUT | 4 - : |
| M - MERGE "DSK | 5 - \NO PREDEFINED |
| N - NEXT | 6 - / VALUE |
| O - OPEN | 7 - : |
| P - PRINT | 8 - : |
| Q - Q | 9 - : |
| R - RUN "DSK | 0 - / |

You can redefine any of these Keys from XBASIC command mode. Letter Keys can be redefined as a macro of up to 16 characters long and number Keys up to 31 characters. You can also save a disk file of your redefined Keys and later load this Key redefinition file into TI-KEYS after TI-KEYS has booted.

AUTOMATIC BOOTING OF YOUR OWN CUSTOM MACROS:

So far so good. But what if you don't like the above standard set of macros that is immediately available when TI-KEYS is loaded? Wouldn't it be nice if you make TI-KEYS automatically set up your own custom CTRL/- macros each time TI-KEYS is loaded, without having to first load TI-KEYS and then load the file that contains your customized macros? EZ-KEYS has this ability. You can modify TI-KEYS to do this using a sector editor.

Use a sector editor that shows the byte number in hex, such as FUNNELWEB's Disk Patch or DISK UTILITIES. The first three sectors of the TI-KEYS LOAD program (sometimes called KEYLOAD, and not to be confused with the D/F80 version of TI-KEYS) contain the ASCII text that is displayed with each CTRL/- keypress. The text for CTRL/A starts at byte >D8 of the first sector, the text for CTRL/B at byte >E8, etc. Examine each of these first three sectors in ASCII and type over any text already there with whatever text you want displayed when you press CTRL/-. Remember, the first 26 macros (letters A-Z) can have 16 characters and the last 10 (digits 1-9,0) can have up to 31 characters. Next move the cursor to the period immediately in front of the ASCII text you just altered (in the first sector byte >D7 for A, byte >E7 for B, etc) and switch the display to Hex (press CTRL/H). Change this two digit hex number to the number of characters in your rewritten text. For example, you might want to change the CTRL/A macro from the original "ACCEPT" (6 characters long) to "ACCEPT AT(" (10 characters long). After changing ACCEPT to ACCEPT AT(in the first sector, move the cursor to byte D7, switch to hex display (CTRL/H) and change the "06" to "0A". Press CTRL/W when done.

The second sector of LOAD (or KEYLOAD) is shown below. It contains text for the macros displayed by CTRL/D thru CTRL/S. You can see that there is plenty of blank space for you to expand or alter these macros. The vertical row of dots is where the hex number is changed.

```

File Editor          EDIT
=====
.  D E L E T E "DSK
.  E N D
.  F O R
.  G O S U B
.  H C H A R C
.  I F
.  J O V S T C
.  K E Y C
.  L I N P U T
.  M E R G E "DSK
.  N E X T
.  O P E N
.  P R I N T
.  Q
.  R U N "D S K
.  S A V E " DSK

```

I have TI-KEYS on my ramdisk as file LOAD. It boots automatically every time I select XBASIC and provides me with the following list of custom macros. Some of these, such as CTRL/4- CALL CAT("DSK1.") are only useful to users of GK-EXTENDED BASIC or the SUPER EXTENDED BASIC (v130) module. Most of the changes from the original predefined macros listed above are usable with any version of XBASIC and are, I believe, real improvements.

```

A - ACCEPT AT(      S - SAVE "DSK
B - BEEP            T - TABC
C - CALL            U - CALL DM
D - DELETE "DSK    V - CALL UCHARC
E - END             W - CALL INIT
F - FOR            X - CALL LOAD("DSK
G - GOSUB           Y - CALL LOAD(-
H - CALL HCHARC    Z - CALL LINKC"
I - IF              1 - RUN
J - CALL JOYSTC    2 - OPEN #1:"PIO" :: PRINT #1:
K - CALL KEYC      3 - OPEN #1:
L - LIST           4 - CALL CAT("DSK1.")
M - MERGE "DSK     5 - LIST "PIO":28:
N - NEXT           6 - LIST "PIO"
O - OPEN           7 - OLD DSK1.
P - PRINT          8 - RUN "DSK1.LOAD"
Q - RUN "DSK4.LOAD" 9 - PRINT #1:
R - RUN "DSK1.     0 -

```

COMBINING TI-KEYS WITH FUNNELWEB:

I mentioned earlier that special possibilities are available if you use TI-KEYS and FUNNELWEB together from a ramdisk. Here is what you can do all automatically just by selecting extended basic from the powerup menu. When you boot TI-KEYS as LOAD from the ramdisk, you can have TI-KEYS automatically boot FUNNELWEB. Many of the TI-KEYS macros are not overwritten by the loading of FWB and are available for use from the FWB XBASIC user list. This means that any XBASIC program you boot from the FWB XBASIC user list can make use of TI-KEYS macros from within the program. Also, from the FWB XBASIC user list you can enter XBASIC command mode by pressing 3 (XB RETURN) and have the macros available. Neat!

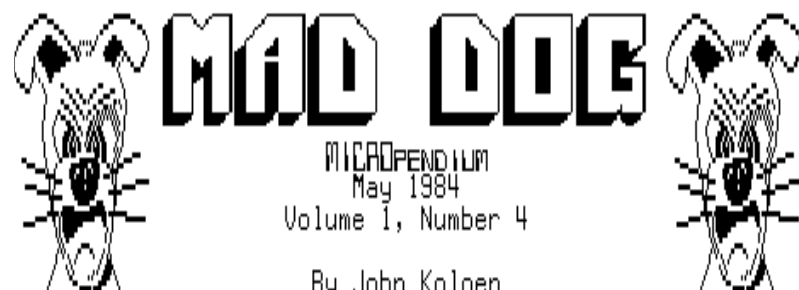
This works with FWB v4.0, v4.1, and v4.11. I haven't tried it with earlier versions. When FWB automatically loads after after TI-KEYS the macros for CTRL/A through CTRL/P are overwritten and no longer available. The remaining macros for CTRL/Q through CTRL/Z and CTRL/1 through CTRL/0 remain available. That is why I have my most used customized macros within these ranges. Here is how to do this. OLD the TI-KEYS LOAD program and list it. There are only 2 lines of XBASIC code embeded in some assembly code. With many hybrid assembly/XBASIC programs you can't alter the XBASIC code and resave the program because doing so messes up the assembly code. TI-KEYS will tolerate slight modifications of its XBASIC code. Change line 20 from CALL LINK("MACRO):: CALL LOAD(-31952,255,231,255,231) to RUN "DSK1.LOADFW" and save this altered TI-KEYS to your

DSK1 configured ramdisk as file name LOAD. Now put FWB's LOAD program on your ramdisk as file name LOADFW. Just select extended basic, and everything boots automatically!

You can't successfully use FWB in automatic combination directly upon selection of XBASIC from the powerup menu with either of the other two "hot keys" programs. If you boot EZ-KEYS or SOFT KEYS and then load FWB, the "hot keys" program won't work.

There is some software that won't load from the TI-KEYS/FWB combination. Some (but not all) XBASIC programs that contain a CALL INIT will lock up the computer if TI-KEYS is already in memory (with or without FWB). These cannot be loaded from the FWB XBASIC user list. An example is the CALL LOAD version of WORD COUNT that was published in Micropendium. To load such programs you first have to turn off TI-KEYS from XBASIC command mode by pressing CTRL/= . Fortunately, there are only a few programs that are incompatible with TI-KEYS. With the TI-KEYS/FWB combination, you can use the XBASIC user list to boot any assembly D/F80 (EA#3) or PROGRAM (EA#5) files, any XBASIC program that doesn't use CALL INIT, and some XBASIC programs that do have a CALL INIT.

TI-KEYS only affects the FWB XBASIC user list. All other parts of FWB, including the other user lists, work normally with the TI-KEYS/FWB combination just as if TI-KEYS was never present.



This is the kind of game that is easy to review. All you have to do is watch the kids play. You know it's not a dud just by counting the number of times they ask to load it into the computer. This game got used plenty the first day, and the next, and the next. You get the picture.

Actually, we're talking about two games here: Mad Dog I and Mad Dog II. The difference in the grades on the report card has to do more with the programming limitations of the Extended BASIC cartridge and the power of the Mini-Memory cartridge than it does with the game itself. Both versions exploit the capabilities of the computer. To avoid confusion, this review will start with Mad Dog I.

Performance: Mad Dog I takes place on a scrolling highway. The player controls a car either by keyboard or joystick input. Appearing randomly on the highway are elusive

mutant spiders which the driver must run over to score points. New ones appear almost as quickly as the old ones are mashed. To make things more interesting, a large truck cruises the highway. Hit the truck and the game is over. There are three levels of play and at each level the roadway becomes narrower. Also, there is a time limit on each level, which increases the challenge.

For an Extended BASIC game, Mad Dog I has excellent sprite coincidence detection. Control of the car is a bit loose, with the car tending to move a bit further laterally than you thought it would. But you can adjust to this rather quickly. Although keyboard input allows you to move the car in only four directions, the joystick option allows eight. However, I found great difficulty in getting this kind of flexible movement out of the TI joysticks. Using a Wico stick made it work as it's supposed to.

Getting to the second level was not particularly difficult. I wasn't able to reach the third level during the time I was evaluating this game. The second level, in addition to taking place on a narrower roadway, introduces green bonus bugs. Hitting these produces extra points. Running off the road results in the deduction of points from your total. The third level includes a second truck and a butterfly which you may run over for points.

MAD DOG II

In a way I wish I hadn't seen Mad Dog II before finishing my review of Mad Dog I. If you've got the Mini-Memory cartridge, by all means spend the extra couple of bucks on Mad Dog II. This is a superior game in every way. Mad Dog II plays like an arcade game. Response to the joystick is instantaneous and flawless.

Mad Dog II is faster than Mad Dog I, has four levels of play and is designed to test the skills of better than average joystick jockeys. This one will give your hand cramps.

The object of Mad Dog II is the same as Mad Dog I. The screens look the same. Unlike Mad Dog I, in which the player has only one car, Mad Dog II provides a spare car upon reaching the third level, no mean feat in itself. This allows you to collide with the truck twice before being eliminated.

Both versions end when time runs out on the final level, though that's a problem few will have to deal with.

Although no big deal is made of sound effects, a tone sounds every time points are scored.

Ease of Use: Both Mad Dog I and Mad Dog II are simple to use. The object of the game is obvious. As far as I was able to tell, both versions are crash proof. Both games are protected.

Documentation: Both games come with a manual that leaves no question unanswered.

Value: If you enjoy arcade-type games, Mad Dog I represents a very good value. The price is excellent for what you get. Mad Dog II is a terrific value for the price. One can spend a lot more and not get a game as good as Mad Dog II. I know I have.



99'er Magazine
February 1983
Volume 2, Number 4
By Steve Schwartz

Why did the chicken cross the road? Most computer games give you the stock reply: "to get to the other side." But the new Chicken game program from Tomputer Software has a much more sensible answer: "... to eat all the chicken feed lying on the highway."

This game is superior to the other "chicken crossing the road" games I've seen. Not only are there good graphics and speed, but you can also move the chicken in eight directions - not just up and down. Pushing the fire button on the joystick will make the chicken quickly jump out of the way of a speeding car or motorcycle.

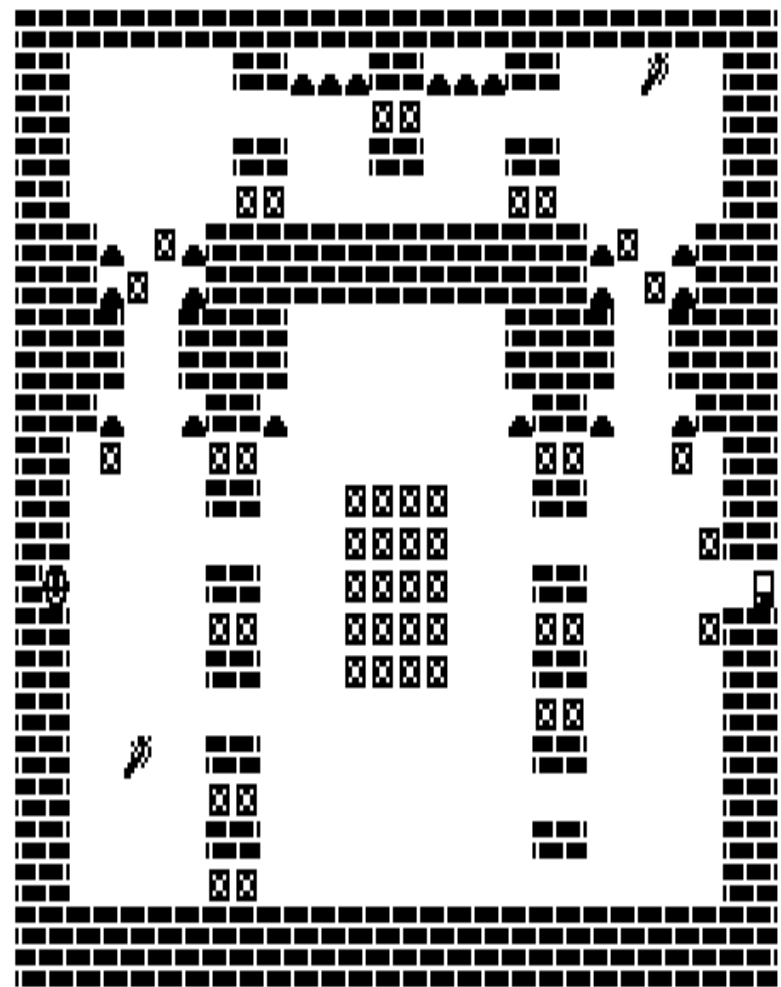
But what makes this game different is that it combines the standard "chicken" game with some elements of Pac-man or Munchman: You are trying to avoid the traffic along this multi-lane highway while at the same time trying to gobble up 10 pieces of corn so you'll progress to the next level of difficulty. Each new level gives you the chance of accumulating more points, but you'll also be facing a greater number of vehicles. And since there are 12 pieces of corn scattered around at each level, you'll have to use some strategy in deciding which ones you can safely eat without being run over.

You start the game with three chickens. At first, the road is very easy to cross. By the time you reach the third level, the highway is bustling with traffic. Make the right moves and you'll be rewarded with bonus chickens. Move too slow and your goose is cooked!

Chicken from Tomputer Software is definitely a game where it pays to eat and run. And \$7.96 is chickenfeed for a game of this caliber.



THE GAUNTLET OF DEATH



By Joe Morris

CONTROL KEYS:

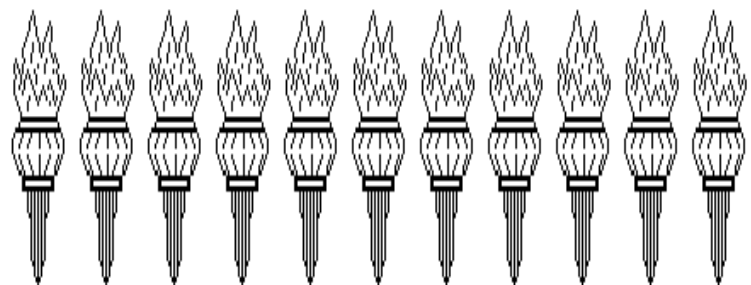
S - Left
D - Right
E - Up
X - Down
SPACE - Use Torch

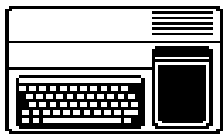
Aim of the Game:

Go through the Gauntlet, get as much treasure as you can, But beware of the invisible traps ...

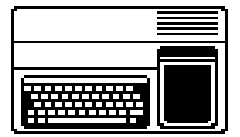
Using the torches reveals the traps for a short amount of time ...

If you lose a life, you lose all treasure you collected





Yesterday's News Information



Yesterday's News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A and Myarc 9640 computers.

TI-99/4A HARDWARE

TI99/4A COMPUTER
MODIFIED PEB
WHT SCSI AND SCSI2SD
MYARC DSDD FDC
MYARC 512K MEMORY
HORIZON 1.5 MEG HRD
TI RS232
CORCOMP TRIPLE TECH
1 360K 5.25 DRIVE
1 360K 3.50 DRIVE
1 720K 5.25 DRIVE
1 720K 3.50 DRIVE

TI-99/4A SOFTWARE

PAGEPRO 99
PAGEPRO COMPOSER
PAGEPRO FX
PAGEPRO HEADLINER
PAGEPRO GOFER
PAGEPRO FLIPPER
PAGEPRO ROTATION
PIXPRO
PICASSO PUBLISHER
BIG TYPE
TI ARTIST PLUS
GIF MANIA

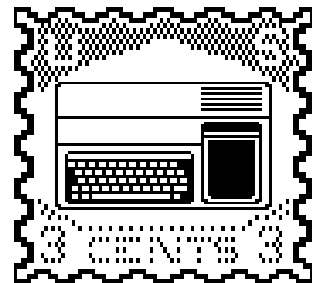
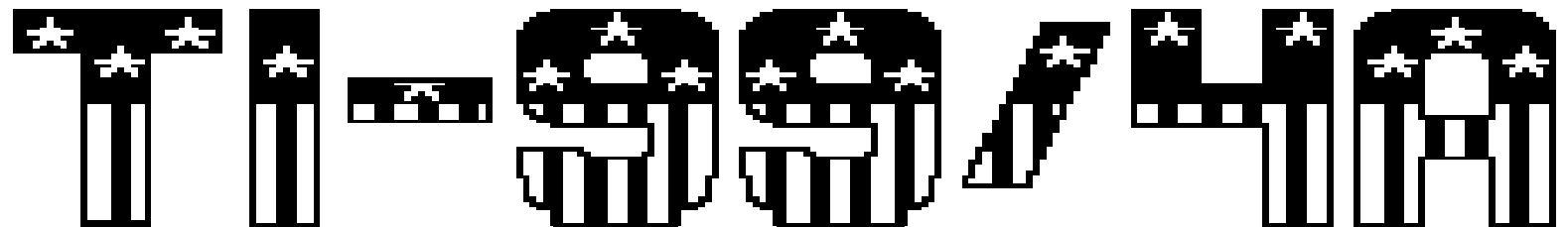
PC HARDWARE

COMPAG ARMADA 2800
COMPAG ARMADASTATION
SAMSUNG SYNCMASTER

PC SOFTWARE

DEAD WINDOWS 98SE
FILECAP
PRNZPBNS
IRFANVIEW
ADOBE DISTILLER
ADOBE AROBAT

Yesterday's News is composed entirely using a TI-99/4A computer system. It consists of 13 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.



TI-99/4A Computer User
1234 What Me Worry Lane
Any City, Any State
Any Country

COMING NEXT MONTH

FILE READER
MAZE MANIA DISK MACHINE
MEXICAN UFOS DODGE 'EM
OSCAR QS SOLITAIRE
GALAXY

I used to have a handle on life, but then it broke.