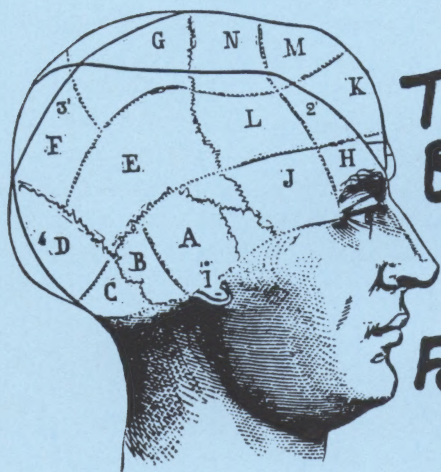
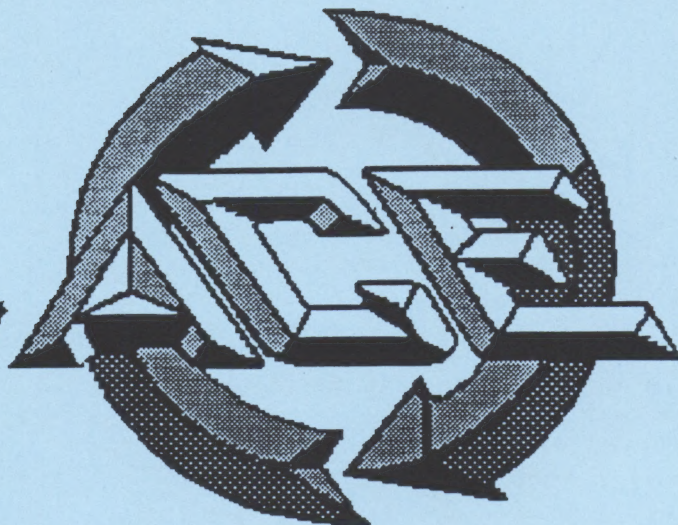


August 1990

Dallas Atari Computer Enthusiasts



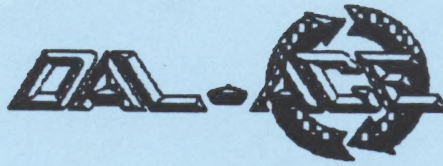
Turbo
Basic
with
Eb
Foerster!

Introduce a
friend to Atari
& Del-Ace!



Whoooo
me?
Back UP!





Editorial Policy

The Dal-ACE-editorial staff reserves the right to edit your submission for spelling, punctuation, grammar, clarity, and for reasons of space limitations.

Disclaimer

The material printed in this newsletter reflects the opinions of the authors. Opposing opinions are solicited. Unless otherwise stated, the material in this newsletter is not copyrighted and no rights are reserved.

The purpose of this newsletter is to present information for your consideration. Neither the editor nor Dal-Ace make claims for the validity or usefulness of this material. The reader is the final judge of any product or advice presented.

... from the 'DOP's' file

Well, last month, in the July issue of DAL-ACE, we made a little mistake. I know you all caught it, but here is the correction anyway. NEW BBS NUMBER * 231-7746 *. Yes, I know, the number I gave you as the new number was the OLD number. What can I say ?

Newsletter Advertisements

Personal ads are free to all current members. Please see the editor for details.

Commercial Rates

Full page \$35

Half page \$25

Quarter page \$15

Business card \$10

For an additional \$10 per full page or \$5 per partial page, you can request that your ad be placed on the inside front or back cover or the center-page spread. This service is first come, first serve.

Ads must be camera ready. Submission deadline for ad copy is the first of the month prior to publication date. That is, November 1 will be the deadline for your ad to appear in the December newsletter. Mail copy to the address on the back page, or contact the advertising manager, editor, or Vice President of Communications. Copy received after the deadline will be run the following month. For contract advertisers, if no new ad is received prior to the deadline, the most recent ad will be run.

Newsletter Submissions

Submissions are welcome in any form. It is requested of any 8-bit user that s/he upload articles to the club BBS or furnish a disk or hard copy to the editor.

Meeting Information

- 10:00 - 11:00 8-bit SIG
- 11:00 - 11:30Disk Sales
- 11:30 - 12:00Main Meeting
- 12:00 - 12:30New users SIG
-Newsletter Exchange SIG
- 12:30 - 2:00.....ST SIG

Infomart Directions

From North Dallas, take either Stemmons (I-35E) or the Dallas North Tollway south. From Stemmons, take the Oak Lawn exit, turn east, and park at the Infomart. If you are using the tollway, exit right on Wycliff, go left on Harry Hines Blvd. to Oak lawn, and turn right. From the south, take Stemmons north, then follow above directions. Infomart is the big, white, steel and glass building south of the other 'marts. GUESTS ARE WELCOME!!!

Sept 15



For 17

TURBO BASIC

An 8-bit Tutorial by Eb Foerster

CORRECTIONS AND CLARIFICATIONS TO DOCUMENTATION FILE.

BLOAD: If the program contains a run address (automatic run with L option of DOS), the command is equivalent to the BRUN command. You cannot load such a file without running it.

DIR: To list a directory of a disk other than drive 1, you must include a file specifier: "Dn:*.*". You cannot use DIR "Dn:".

EXIT: Documentation states "Exit a DO-LOOP loop". This command exits all loops, DO-LOOP, REPEAT-UNTIL, WHILE-WEND, and FOR-NEXT. A PROCEDURE may also be exited with this command. I would advise against using it for this purpose. The EXIT goes to the line after the ENDPROC command. It does not return to the line calling the procedure.

PAUSE n: The actual pause is n/60 seconds. To pause 5 seconds, you would program PAUSE 300.

RENUM: Be careful with this command. It is possible to have duplicate line numbers or create sequences of lines that are not in numerical order.

RAND(n): Generates a random integer in the range from 0 to n-1.

INSTR: If you use more than one INSTR function of the short form - INSTR(X\$,A\$) - in an expression, you will get "ERROR- 3 VALUE". The long form - INSTR(X\$,A\$,i) - returns position within X\$, not relative to start position i. The long form also starts search at i+1. INSTR("ATARI", "A", i) returns 3 if i=1, 0 if i=3, and 1 if i=0.

UINSTR: Same rules as for INSTR above apply.

ERROR-27: Occurs when program flow encounters a PROC. You must enter a procedure with the command EXEC.

IF-ELSE-ENDIF: If you use an implied GOTO within this structure, you will get an "ERROR-22 ?NEST":

```
IF X > 10
  IF Y = 5 THEN 2500
ENDIF
```

These lines will execute if the condition statements are true, but will return "ERROR 22" if one of the conditions is false. You must use the explicit GOTO statement:

```
IF X > 10
  IF Y = 5 THEN GOTO 2500
ENDIF
```

The IF-ELSE-ENDIF is designed for multi-line statements within the IF structure. However, this form of the statement may also be written on one line:

```
IF CHR$(A)="Y": PRINT "YES": ELSE:
PRINT "NO": ENDIF
```

RUNNING ATARI BASIC PROGRAMS ON TURBO BASIC

Turbo BASIC will run most Atari BASIC programs as is without modifications. You may, however, have trouble with some programs because the speed of execution is much faster than in Atari BASIC.

An example is the use of an empty FOR-NEXT loop in Atari BASIC for timing a delay (FOR A=1 TO 100:NEXT A). Such loops are used to time note durations or to display a screen for a specified period of time. These loops will execute 3 times faster in Turbo BASIC and 5 times faster when run as a compiled program. The solution to this problem is to change all empty timing loops to PAUSE statements. The time delay will then be exact whether it is run with Turbo BASIC or as a Compiled program.

One other problem you might encounter is the use of more than one NEXT statement in a FOR-NEXT loop:

```
100 FOR A=1 TO 5
110 IF A=3 THEN NEXT A
120 ? A
130 NEXT A
```

If you try to compile these lines, you will get an error at line "130: No FOR". In other words you must have one and only one NEXT statement for each FOR statement. To compile the above lines, you would need to change the "NEXT A" in line 110 to "GOTO 130" or change the lines using IF-ELSE-ENDIF branching.

I have discovered these errors in documentation and restrictions in the use of the new commands. If you discover any others, please send them to me and I will pass them on in future articles.

INDEXED DATA STATEMENTS

Last month, we presented a program which, when run, printed a message to the screen. It was not obvious from the program what the message would be. If you figured it out before running the program, congratulations. The program demonstrated how you can manipulate the pointer that Atari BASIC uses as an index to an item within a DATA statement. This pointer is at memory location 182. You can make use of this pointer to read specific items from a line of DATA. The example last week was kind of obscure and of no practical value unless you like writing programs that are difficult to decipher. Let's look at another example:

```
100 DIM A$(10)
110 CLS
120 PRINT "PRESS ANY NUMBERED KEY"
130 DO
140 GET A
150 IF A>47 AND A<58
160 RESTORE 220
170 POKE 182,A-48
180 READ A$
190 ? A$
200 ENDIF
210 LOOP
```



220 DATA ZERO ,ONE ,TWO ,THREE ,FOUR
,FIVE ,SIX ,SEVEN ,EIGHT ,NINE

The DO-LOOP at lines 130-210 is an infinite loop. The GET command at line 140 waits for a key to be pressed and places the ASCII value of that key in the variable A. Line 150 tests if the key is a number 0-9 (ASCII value 48-57). If true, the statements from 160 to 190 are executed, else the program loops. Line 160 makes sure that you are pointing to the desired line containing the DATA. Line 170 then pokes a value from 0 to 9 into memory location 192. The READ statement then reads the corresponding alphabetic representation of the number and prints it.

Using the RESTORE command, you can only point to a line containing a DATA statement. Using location 182, you can now point to a specific item within a DATA statement. In other words, the use of this pointer allows you to use the DATA statement as a look-up table.

Here is another program using the indexed DATA statement. This one, however has more practical value. It will test whether your joysticks are are functioning properly.

```
100 GRAPHICS 23
110 COLOR 1
120 X1=0:Y1=0
130 PLOT 80,40
140 DO
150 A=STICK(0)
160 RESTORE 290
170 POKE 182,A
180 READ X
190 RESTORE 300
200 POKE 182,A
210 READ Y
220 IF X<>X1 OR Y<>Y1
230 COLOR 0:DRAWTO 80,40
240 COLOR 1:DRAWTO
80+30*X,40+30*Y
250 X1=X:Y1=Y
260 ENDIF
270 IF NOT STRIG(0): TEXT 35, 5, CHR$
(20): ELSE :TEXT 35,5," ":ENDIF
280 LOOP
290 DATA ,,,,1,1,1,-1,-1,-1,0,0,0
300 DATA ,,,,1,-1,0,,1,-1,0,,1,-1,0
```

PROGRAMMING PROBLEM

Here is an interesting problem for you to solve on your computer, whether you are a beginning or an experienced programmer. The NCAA recently completed their basketball tournament. Sixty-four teams were entered and played in the single elimination tournament (losers went home; winners went on to the next round). Write a program that will calculate the total number of games played in the tournament.

If you are a really good programmer, have your program calculate the number of games played in a single elimination tournament where the total number of teams are not a power of two (2,4,8,16...). In other words, some teams will have a first round bye. Give it

a try; we will present a solution for you next month.

Next month we will start examining some of the new commands in more detail.

DOLLAR SIGNS

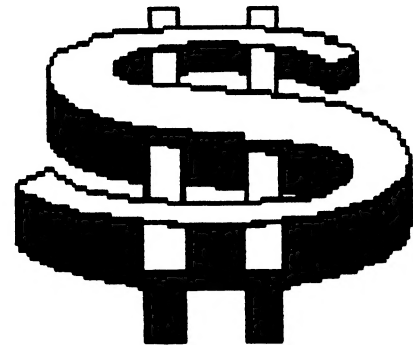
Financial Reports for the months of June and July are as follows:

JUNE

Income \$26.00

Expenses -0-

Gain(Loss) \$26.00



JULY

Income \$9.00

Expenses \$157.49

Gain(Loss) (\$148.49)

Turbo Basic!
Now at your
disk library
82!



MegaByte Computers

ST

of
North Texas
909 Melbourne
Hurst, TX 76053
Metro (817)589-2950

Notice
We would like to remind you of our policy concerning software reviews. If you would like to review a piece of software for the club, drop by and we will loan it to you for one month. Call the store for more details.

"Where can I find ANYTHING for my computer?"

Are you an 8-bit Atari owner who is frustrated by the lack of support for your computer? We at MegaByte understand your feelings. We started our computer experiences with the 8-bit Atari systems too. Our earliest computer sales experience was with the 8-bit Atari systems, as well as our early technical experiences. In spite of this, the sad reality is that new Atari buyers are looking for the ST, Mega, Stacy, and Portfolio products. So, does this mean that 8-bit Atari owners are just out of luck? We don't believe so! We believe that people make their own luck. That's why MegaByte Computers is continuing with its offer to 8-bit Atari users that everytime you order a program that we do not have in stock, we will order a second program to put on our shelves. As a matter of fact, we'll let YOU decide which program. If you select a program that we have in stock, we will reorder that title (if available) as well as a second program for stock (again, your choice.)

At MegaByte we believe that you want to buy for your computer locally, and will do so if given a chance. This policy will give you, the end user, an opportunity to show us, as well as the other area Atari dealers, the viability of your machines.

We also believe you will be suprised at the number of software titles that we currently stock. We now have approximately 100 8-bit titles in stock, as well as several different pieces of hardware. We have several pieces of used equipment as well, all at extremely low prices.

Please, come by our store and let us know what you want us to carry for you computer system. We are anxious to supply your computer needs, and we believe that the users in this area are anxious for a dealer to do so!

MegaByte is proud to endorse The Revolution
Join The Revolution...Use an ATARI computer

User's Group special - We are offering the new Zoom/Modem 2400 to members of any of the local Atari user's groups for \$124.97. This is your chance to go on-line at 2400 baud for a very low price. A valid user's group ID card is required for this price.

Also Included - ONE WEEK OF STARTEXT FREE!

MegaByte Computers

Now you have a choice

Copyright and Bankruptcy

by Sandra Higgins Hanna

If you read ANY other computer mags at all, you have read of the upcoming turmoil about the copyright law. According to the Council of European Communities, it will be an infringement of copyright to "make, import, possess or deal with articles intended to facilitate the removal or circumvention of any technical means which may have been applied to protect the program". What does this mean to us Atari guys 'n dolls, well, it means that we, under law, will not be able to make backup copies, or copies of defunct software companies that leave no forwarding addresses. It means, if you take the term "technical" to mean that little piece of foiled adhesive paper they put over the notch that we all remove to make those nasty ole backups and/or copies, will hitherto be unlawful and punishable. Britain has already passed such a law, adding to it that it is unlawful even to discuss copy-protection bypasses! How 'bout them apples! What about the laptop issue? Does that mean that your software residing on your mother Atari, cannot share with its child, the Stacy, the Portfolio, etc, ad infinitum? You betcha. And what, specifically, is the 8-biter to do when his/her favorite software company cum game programmer bites the dust and now lives in a hut in east Wazoo? Huh? We can't even write the guy to buy him out, or work a deal to make copies of his programs floating among us and send the guy a check every month or so. It's always somethin' ain't it? Well, now that I have you hot under the collar let me divulge the had guys. Please boo at will!

Software Publishing Association * The Business Software Association * Apple
 What are we to do? Write a steaming letter to these guys, explaining in gentle manner, our problem concerning defunct 8-bit software companies, backups, mother-child sharing, whichever is your gripe. By now, you should have one. //
 //

SOUTH

Huntsville, Alabama
Atari Users Group

TURBOBASIC: NOW A NORTH AMERICAN VERSION!

By John Picken

Garden City ACE (GCACE) British Columbia, Canada. (Reprinted from March 1989 Puget Sound Atari News). Recently PSAN readers became the first, if not the only Atarians, who can use TextPro with Mydos. This month, those of you willing to suffer through the typing can be the first on your block (your continent actually) to use the TIMES\$ functions of TurboBASIC.

Listing 1 follows this article will accomplish the following:

- A. "Fix" TurboBASIC so that time functions work correctly on North American Ataris.
- B. Change the existing autorun filespec from "D:AUTORUN.BAS" to "D.?????.ARB".
- C. Change the prompt from READY to TURBO to identify the revised version.
- D. Optionally, allow you to vector anywhere you want with the BYE command.

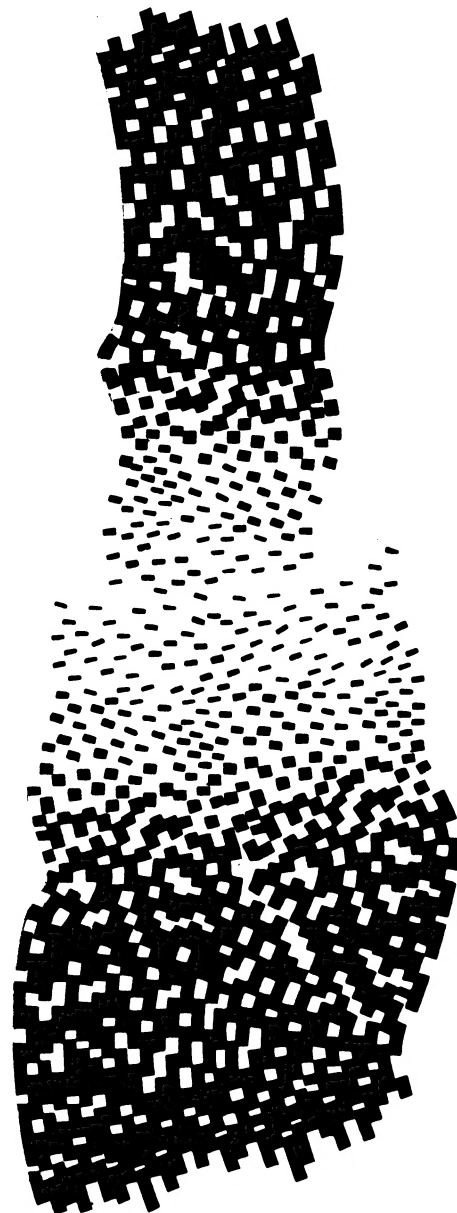
WHY THE TUROBASIC TIME COMMANDS DON'T WORK ON NORTH AMERICAN ATARIS.

The reason why TurboBASIC's TIMES\$ command doesn't work on our variety of Atari lies in the frequency of NMI's. (Non-Maskable Interrupts - periods when the screen gets updated and other "housekeeping" operations are carried out). This frequency is the same, not coincidentally, as the frequency with which your television or monitor redraws its display, and the

same, also not coincidentally, as the AC power frequency.

One of the things that happens during an NMI is an increment of the memory locations known as the Real Time Clock. Despite its impressive sounding name, this is nothing more than three bytes of RAM. In other words, all it holds is bits - no minutes or seconds. So these bits have to be converted by calculation into time. Since the clock is incremented once every NMI, all we have to do is take the total value in the clock locations and divide by the NMI frequency to know how many seconds have elapsed since time zero.

Ataris sold in Europe, where TurboBASIC originated, behave exactly the same way as ours. The crunch is that the frequency is



different. European (PAL) Ataris have 50 NMIS per second while our (NTSC) models have 60. This means the division mentioned above produces the wrong result.

HOW THE MODIFICATION PROGRAM WORKS.

Listing 1 fixes TurboBASIC in two spots. First, it replaces a value, in floating point format, for total NMIS per day with a value based on our 60 Hz standard. That's all that's needed to get the TIME\$ function to report correctly. To correct the set time feature, the program finds a pair of subroutine calls that look something like this:

```
GOSUB MULTIPLY_BY_5  
GOSUB MULTIPLY_BY_10
```

and replaces them with a single

```
GOSUB MULTIPLY_BY_60
```

Since you can't just chop bytes out of a program and expect it to run correctly, Listing 1 replaces that second subroutine call with NOPS (No Operations - 6502 machine language instructions which do nothing).

The other changes Listing 1 makes are primarily convenience features. The AUTORUN filespec is changed from "D:AUTORUN.BAS" to "D:??????.ARB" which means you can have several potential AUTORUN files on the same disk and know by its name what each one does. The one which will actually run is the first one in the directory with the "ARB" (Autorunbasic) extender. Listing 1 contains REMS to assist you in changing this filespec.

Listing 1 also changes the prompt from "READY" to "TURBO" to let you know which version you're using. This has the side benefit of letting you know TurboBASIC survived your last RESET. (It doesn't always, depending on the DOS you're using and whether or not you're using MEM.SAV.)

The final change made by Listing 1 is optional and is included for advanced users. If you don't want it, or don't understand it, just ignore it. All it does is give you a hook into machine language without the requirements to do a PLA, as you have to when you use USR. The new hook is used by DPOKEING the address of the machine language subroutine into \$0100 and jumping to the subroutine

via BYE. The new hook is not all that useful since your M/L routine must end with a JMP -- it can't RTS so it can't be used except as a terminal instruction in a BASIC program.

RUNNING LISTING 1.

First make a copy of TurboBASIC and store your original copy of TurboBASIC away for safe keeping. Next type in Listing 1. Check it for errors and SAVE Listing 1 to disk. When you're sure Listing 1 is typed in correctly RUN it. (Note that Listing 1 will only operate correctly when RUN from TurboBASIC, and not from Atari BASIC.) Follow the prompts to customize TurboBASIC to your liking.

TURBO TIME FUNCTIONS - "DOC's"

PAUSE does exactly that. It pauses until the Real Time Clock has increased by the specified number of counts ("jiffies"). PAUSE 60 (which would be 60/60 of a second) is analogous to:

```
10 X=60+PEEK(20)+256*PEEK(19)  
20 Y=PEEK(20)+256*PEEK(19)  
30 IF X>Y THEN GOTO 20
```

TIME is the sum of the "jiffy" count in the Real Time Clock. The statement X=TIME produces identical results to this:

```
10 X=256*256*PEEK(16)  
20 X=X+256*PEEK(19)  
30 X=X+PEEK(20)
```

TIME\$ is used in two ways:

PRINT TIME\$ or A\$=TIME\$ is how you get a report of the time of day. This report is in the form hhmmss (six digits) where "hh" refers to hours based on the 24 hour clock, "mm" is minutes, and "ss" is seconds. The second way TIME\$ is used is to set the clock. This can be done either directly using TIME\$="hhmmss" or via a string equate using TIME\$=A\$.

To understand how TIME\$ works check the following where "J" represents total "jiffies", "NMI" is the frequency of the nonmaskable interrupts (60 per second for us, 50 per second for Europe) and "hhmmss" are six digits as above. The statement TIME\$="hhmmss" has the same effect

as:
10 J=NMI*(SS+MM*60+HH*60*60)
20 POKE 18,(J DIV 65536)
30 POKE 19,(J MOD 65536) DIV 256
40 POKR 20,(J MOD 65536) MOD 256

There are some peculiarities. Though there's a dollar sign in the name, TIME\$ is not a string and not a string operator. If you want to use something like A\$=TIME\$, then A\$ must be DIMENSIONED as normal but TIME\$ is not.

Also, you can't use string operators on TIME\$. If you want the equivalent to TIME{1,4} you have to assign it to a string and then work on that string. Finally, when setting time, if the value for "hh" is greater than 23 or the value for "mm" or "ss" is greater than 59, or if any of the si figures are non-numeric, you get an error condition which can be TRAPPED and the time is not changed. (See line 170 in the LISTING 2.)

USING IT - TWO SHORT EXAMPLES

EXAMPLE 1 -- LISTING 2: I use Listing 2 as a nine line header when developing a TurboBASIC program. It approximates an autosave feature. To use it you first set the clock using the TIME\$ statement. Then either RUN it or, if you want to start LISTING at a later line number, specify a value for L and then GOTO 1. For example:

```
L=2000:G. 1.
```

The header will renumber your program, LIST it from line L on, SAVE it to the RAMdisk, show you its sector length, and STOP. If that's all you want to do, then press BREAK. If you want to make a copy to a floppy, then hit RETURN, it will SAVE the program (including the header) to D1: using a filename consisting of four letters plus the hour and minutes. The extender will be a version number used in conjunction with the variable M. ("M" stands for maximum number of versions to be kept on the disk.) Again, if this is all you want to do, then press BREAK to exit. If you want to save the final version of the program to floppy, without the header, then press RETURN and the header will strip itself from the program, write the program to floppy, reset the version number and end on an error 15. (The

normal mode of ending from the last prompt.)

Adjust lines 1 and 2 to suit yourself; i.e. the RENUM values, your RAMdisk number, a four letter filename, and a value for M. Note that M must not be 0 or what is SAVED will be immediately erased. Note also, in lines 4 and 7, the "+++" should be typed in as "up arrows; i.e., [ESC],[CONTROL],[-].

EXAMPLE 2 -- LISTING 3: The Real Time Clock is fairly accurate. The SpartaDOS doc's claim it's good to within a minute per day and, while SpartaDOS' calendar/clock functions are more complex, they are dependant on the internal clock (unless you use R-Time8 which is a whole different ball of wax). In practice I've found the accuracy to be at least that good as long as you don't carry out alot of I/O. Once you start sending or receiving stuff via the serial bus (diskm cassette, printer, etc.) the clock slows down, if you use Spartads, then you've already seen the effect. SIO is almost entirely interrupt driven and, I suspect that the occurrences of VBI's or "Vertical Blank Interrupts" (NMI's when the clock is incremented) are delayed by the SIO interrupts. Books I've found on the subject indicate that other IRQ's or "maskable" interrupts are disabled when the 6502 is interrupted, but the slow down of the clock would seem to indicate that all interrupts are masked by the occurrence of any one. Whatever the cause, if you start doing serial I/O, "real" is no longer really "real".

Listing 3 is an historical first -- the first program I've written in three years that departs from GRAPHICS 0. (My XE and XL both believe that, graphically speaking, they are "TRS-6502s").

Anyway, if you want to try it out, the following listing turns your computer into a digital clock. It's short and simple, meant only to demonstrate the use of the TIMES\$ statement. If you type it in, be careful of the punctuation in the PRINT lines. When you run it, you'll get some indication of the accuracy possible and also the proff that my fix for TurboBASIC's time statements really works. Enjoy.

(ED HAUG - In the following code examples I have seperated each code line with a blank line for clarity.)

LISTING 1

```
1 REM TurboBASIC Modifier jkp
12/89
2 REM
3 REM Changes TurboBASIC file on
4 REM disk as follows:
5 REM a. adjusts TIME$ function to
6 REM work with NTSC Ataris
7 REM b. changes prompt to
"turbo"
8 REM to identify version
9 REM c. changes autorun filespec to
10 REM D:??????.ARB
11 REM d. (optional) changes BYE to
12 REM indirect jump through $0100
13 REM -----
20M=19000:DIMT{M},F{41},N{13},S
{41}:T{M}=CHR{0}:POKE 82,8:
POKE 694,0:POKE 702,64:CLS
:PRINT
30 REM -----
40 N$="D:??????.ARB"
50 REM New autorun filespec. You
can make it "D8:??????.ARB,etc. If
you don't want to make any change
then
60 REM make N$="D:
AUTORUN.BAS"
70 REM -----
80 PRINT " Turbo-BASIC Modifier"
:PRINT :PRINT " courtesy of
the":PRINT :PRINT "Puget Sound
Atari News":PRINT
90 PRINT "CAUTION - Work on a
copy,":PRINT "this program is
FINAL.":PRINT "It won't work on
files"
100 PRINT "once they are
modified":PRINT "(Including this
mod.l)"
110 PRINT :PRINT "Insert
```

```
TurboBASIC disk":PRINT "Enter
complete filespec":PRINT :INPUT
"=> ",F$
120 REM -----
130 TRAP 810:OPEN #1,4,0,F$
140 TRAP 150:BGET #1,ADR(T),M
150 CLOSE :T$=T{1,DPEEK($0358))
160 REM -----
170 FOR X=1 TO 6:READ
Y:S{X}=CHR{Y}:NEXT X:DATA
67,4,50,0,0,0
180 X=INSTR(T$,S):ON X=0 GOTO
800
190 T{X+1,X+1}=CHR{5}
200 T{X+2,X+2}=CHR{24}
210 T{X+3,X+3}=CHR{64}
220 REM
230 FOR X=1 TO 6:READ
Y:S{X}=CHR{Y}:NEXT X:DATA
32,242,251,32,236,251
240 X=INSTR(T$,S):ON X=0 GOTO
800
250 FOR X=X TO X+5:READ
Y:T{X,X}=CHR{Y}:NEXT X:DATA
32,19,252,234,234,234
260 REM -----
270 X=INSTR(T$,"YDAER"):ON
X=0 GOTO 800
280 T{X,X+4}="OBRUT"
290 REM -----
300X=INSTR(T$,"D:AUTORUN.B
AS"):ON X=0 GOTO 800
310 T{X,X+12}=N$
390 REM -----
400 PRINT :PRINT "Turbo-BASIC
patched":PRINT "for NTSC,
AUTORUN":PRINT "filespec
changed."
410 PRINT :PRINT "OPTION to write
file"
420 PRINT :PRINT "START to
```



```

change BYE":PRINT CHR{28};
430 R=PEEK(53279):ON R=3 GOTO
690:ON R<6 GOTO 430
440 REM -----
450 REM START
460 PRINT :PRINT :PRINT "If you
don't understand"
470 PRINT "the implications of the"
480 PRINT "following change, don't"
490 PRINT "make it - press OPTION"
500 PRINT "to write the file as is."
510 PRINT
520 PRINT "The JMP $E471
accessed"
530 PRINT "via the BYE statement is"
540 PRINT "replaced by JMP
($0100)"
550 PRINT "and the file is changed"
560 PRINT "to load $E471 into
$0100"
570 PRINT :PRINT :PRINT
580 PRINT "OPTION to write file"
590 PRINT :PRINT "START to
change BYE":PRINT " and write
file":PRINT CHR{28};CHR{28};
600 R=PEEK(53279):ON R=3 GOTO
690:ON R<6 GOTO 600
610 REM -----
620 FOR X=1 TO 3:READ
Y:S{X}=CHR{Y}:NEXT X:DATA
76,113,228
630 X=INSTR(T$,S):ON X=0 GOTO
800
640 T{X,X}=CHR{108}
650 T{X+1,X+1}=CHR{0}
660 T{X+2,X+2}=CHR{1}
670 T$=T{3}
680 REM -----
690 S$="":PRINT :PRINT :PRINT
"Ready to write file":PRINT:PRINT

```

```

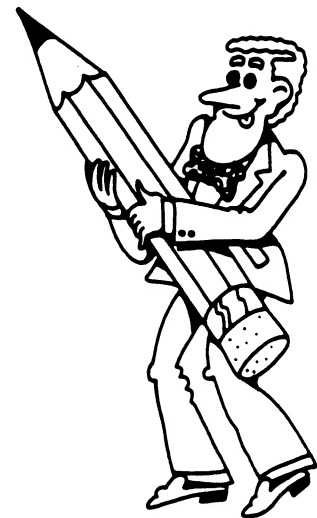
"Enter new filespec or"
700 PRINT "(caution) to write
to":PRINT F$:PRINT
710 PRINT CHR{156};" just hit
RETURN":INPUT "=> ",S$:IF S$=""
THEN S$=F$
720 PRINT S$:F$=S$:TRAP
730:UNLOCK S$
730 TRAP 810:OPEN #1,8,0,S$
740 ON R<>6 GOTO 750:PUT
#1,255,255,0,1,1,1,113,228
750 BPU #1,ADR(T),LEN(T)
:CLOSE
760 PRINT :PRINT "Another copy
OPTION":PRINT :PRINT "Quit
program START";
770 M=PEEK(53279):ON M=3
GOTO 710:ON M<6 GOTO 770
780 POKE 82,2:PRINT :TRAP
40000:END
790 REM -----Error handlers-----
800 PRINT "File previously
modified":GOTO 780
810 CLOSE :PRINT :PRINT "DISK
ERROR #":ERR:PRINT :ON S$=""
GOTO 110:GOTO 760
LISTING 2
1 RENUM 10,10,10:M=3:CLS :TRAP
2:DIM F{41}
2 F$="D4:WORK":TRAP 40000:LIST
L+10*(L<10),:SAVE F$:PRINT :DIR
F$
3 V=PEEK(1536):F{2,2}="1":F{8}
=TIMES:F{12}="":F{13}=STR{V}
4 PRINT "CONT:REM save to
D1:++":STOP
5 TRAP 9:PRINT "Saving ";F$:POKE
1536,V+1:SAVE F$:ON V<M GOTO
7:F{8}
="*":F{10}=STR{V-M}
6 ? "Erasing ";F$:DELETE F$
7 PRINT :F{8}="*.*":DIR F$:PRINT
"CONT:REM Del 0-9/reset

```

```

version.++":TRAP 40000:STOP
8 POKE 1536,0:DEL 1,9
9 PRINT "DISK ERROR ",ERR:END
LISTING 3
10 GRAPHICS 0
20 DIM H{6}
30 PRINT
40 PRINT
50 PRINT
60 PRINT "Enter a new time or"
70 PRINT "RETURN if no change"
80 PRINT
90 PRINT CHR{127};CHR{127};
100 PRINT " hhmms";CHR{157};
110 PRINT CHR{127};CHR{127};
120 PRINT "=> ";TIMES;
130 FOR X=1 TO 6
140 PRINT CHR{30};
150 NEXT X
160 INPUT #16,H$
170 TRAP 190
180 TIMES= H$
190 GRAPHICS 2+16
200 DO
210 H$=TIMES$
220 POSITION 6,5
230 PRINT #6;H{1,2};
240 PRINT #6;":";
250 PRINT #6;H{3,4};
260 PRINT #6;":";
270 PRINT #6;H{5}
280 LOOP

```



Nightmare on Lawrence St. Or Backups? I Don't Need No Stinking Backups!

Gregg Anderson (RACE)

Well, it finally happened to me. After two years harping to others on the importance of keeping a current backup of their hard drive, my system went down and took over 40 Megabytes of data with it.

No problem, you say? Just re-format and load in your last backup, you say? After all, you've been around long enough to know how important backing up your hard disk is. You're always telling us to do it, so you must be doing it yourself. Right?

Wrong!!! In the famous tradition of "do as I say, not as I do," I kept delaying my backup. After all, it takes hours and hours to do a floppy backup of a 40 meg hard disk and I just didn't have the time. I was too far behind on my articles and had a bunch of fonts to modify before I could take the time for a backup. So off I went, working on four different articles, modifying *lots* of fonts for LaserBrain, beta testing some new utilities, and generally staying pretty busy.

Then about two weeks ago, I finally got around to installing a new hardware modification I'd been promising myself for months. It was just a simple 'plug in' job and my ST had a similar unit installed a few months ago for a review (and removed after finishing the article).

The upgrade seemed to be working all right so I pressed on and hit the old WordWriter ST again. All was going great until I tried to print something and found the text was out of sequence. Since the screen display looked normal, I

re-saved the file and printed it again.

This time everything looked ok, until I read the file and found the system had somehow deleted a paragraph and substituted a paragraph from a totally different article. Now, I may be a little slow from time to time but even I'm bright enough to see I had a problem here. I decided to re-boot and start over—bad choice!

From that moment on my system no longer accessed the hard drive. In fact, it couldn't even tell a hard drive was connected. After sitting on the panic button for a while, I finally decided to bite the bullet and reformat the disk.

That was when I remembered how out of date my backup was. ARG!!! All the articles I'd been working on, all the fonts I'd modified, *everything* I had done was gone and would have to be re-created from scratch. After kicking myself repeatedly, I limped over to the computer and got started.

I reformatted and partitioned the beast and everything seemed all right. Until it scrambled itself again half way through the reload, that is. At this point I removed the upgrade (at the suggestion of the manufacturer) and started over. This time everything worked and has continued to work without so much as a hiccup from the drive.

So what happened? As near as we can tell something in the modification imposed a load on the disk controller, the DMA/MMU chips, or on the 8 Mhz clock that wasn't there before. Somehow this allowed WordWriter to partially overwrite the disk's directory and File Allocation Tables (FATs).

This shouldn't have happened

and is the only such incident the company knows about (they want to borrow my system and find out exactly what happened to keep it from happening again; this outfit cares about its product and customers). So don't let my disaster keep you from upgrading your ST, but do learn from my biggest mistake.

Me admit to making a mistake? As much as it hurts to admit it, I'm afraid so. *I didn't have a current backup of my hard disk!!!!* By trying to save a few hours of time I lost more than three months worth of hard work, quite a few dollars in phone calls, and at least two inches off my hairline.

Always do a complete backup before starting any type of modification to your system. Even if you know for a fact what you're doing can't possibly cause problems with your hard disk, *back it up first!*

If you can't afford a dedicated backup utility like Seymour-Radix's VCR tape backup, then look into The Vault. The Vault is available on this month's AIM Disk of the Month, or for download on all the major on-line services, and does a fine job of backing up to floppy disks any hard disk out there. The Vault is fairly quick, for floppy backup, and will even work smoothly in only a 512K machine.

It all boils down to this; Save yourself a LOT of excess worry, unplanned work, heartache, and maybe even some money. If you've not backed up your hard disk lately, DO IT NOW! And enjoy a little peace of mind.

I'm a bit sadder and balder, but definately wiser!



DAL-ACE VOLUNTEERS

Editor:

Sandra Hanna 262-0196

Ad Manager:

Marc Salas 717-4615

BBS Sysop:

Jim Jackson 264-0205

8-Bit Librarian:

Tim Mixon 356-4725

ST Librarian:

OPEN

DAL-ACE OFFICERS

President:

Donny Arnold 216-0766

Vice President:

Dave Gramm (214)370-7143

Secretary: Harold Lewis 298-1734

Treasurer:

Rene Tucker 223-6176

Vice President of Communications:

Sandra Hanna 262-0196

Members at Large:

Billy McLean 285-3806

Harold Lewis

Tim Mixon 356-4725

DAL-ACE BULLETIN BOARD

(214)278-6180

Dal-Ace Experts

Donny Arnold 289-6746 call before 10 p.m.

8-bit general knowledge

Eb Foerster 357-7602 call from 7 to 10 p.m.

Turbo BASIC, SynFile, SynCalc, Assembly

Ron King (817)283-0674 call from 5 to 10 p.m.

8-bit hardware

John Saunders (817)566-0318

C and Assembler languages

Ralph Tenny 235-4035 call from 7 to 10 p.m.

ST general knowledge and hardware

Rene Tucker 223-6176

8-bit general knowledge

DAL-ACE INC.

Dallas Atari Computer Enthusiasts

DAL-ACE is an independent user education group that is not affiliated with the Atari Corporation. This is the official newsletter of DAL-ACE and is intended for the education of its membership as well as for the dissemination of information about Atari Computer Products.

DAL-ACE membership is \$20 per year. Lifetime Membership is \$250. BBS Only membership is \$10 per year. This newsletter is written, edited, and published by volunteers. Its availability and/or distribution may, at times, be subject to circumstances beyond the control of the club officers. A pink address label indicates that your membership expires this month. Other user groups may obtain a copy of this newsletter on an exchange basis.

ALL MATERIAL PRINTED IN THIS NEWSLETTER MAY BE REPRINTED IN ANY FORM PROVIDED THAT DAL-ACE AND THE AUTHOR, IF APPLICABLE, ARE GIVEN THE PROPER CREDIT. LIKEWISE, PORTIONS OF THIS NEWSLETTER MAY BE REPRINTED FROM OTHER NEWSLETTERS AND ARE SO NOTED.

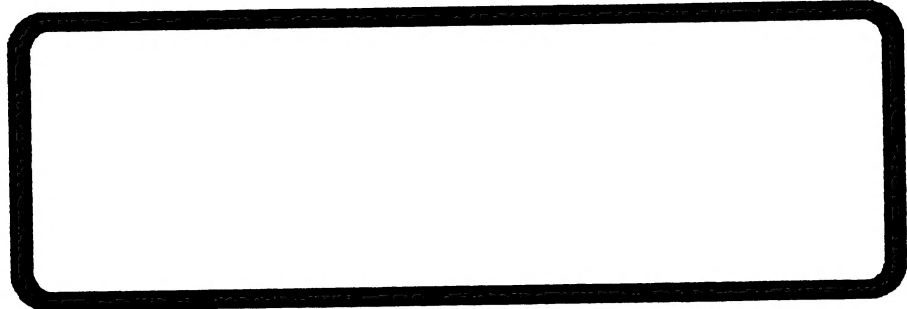
Volunteer—
help make
your club
a better
place to
meet.

Write an
Article
For
Your
Club
Today

*June BB
attn
231-7746*



P.O. BOX 851872
RICHARDSON, TEXAS 75085-1872



Postmaster: Return Postage Guaranteed. Address correction requested.

Is it that time again? If your renewal number has been circled (above), use this handy form to keep your DALACE Newsletter coming!

NAME _____

ADDRESS _____ APT. # _____

CITY _____ STATE _____ ZIP _____

PHONE # HOME _____ WORK _____

HOW YOU FOUND OUT ABOUT DAL-ACE:
PLEASE CHECK ONE

DAL-ACE MEMBER * _____ FRIEND _____ STORE* _____

OTHER _____ *GIVE NAME _____

USER # _____ PASSWORD _____

Make your \$20 check to: DAL-ACE

Mail to: DAL-ACE

**P.O. BOX 851872
RICHARDSON, TEXAS 75085-1872**