

TEXAS INSTRUMENTS

TI Forum

by Ron Albright
and Jonathan Zittrain

Watford Introduces A New Terminal Program: Omega

Travis Watford, the programmer responsible for the Max-RLE graphics viewing program, has released yet another project to the public domain. It is a terminal emulator program that supports on-line RLE graphics viewing, programmable keys, one-time default setup, and XModem file transfers.

One of the most useful features of Omega is its buffer mode. After pressing <FCTN>B, a user can enter information on the bottom few lines of the screen while output is placed on the rest of the screen. This buffer makes on-line conferencing much more efficient, since one's typing is not interrupted by incoming messages. Transcripts made by a conference participant using the <FCTN>B option also come out much cleaner.

Other current functions include a word wrap mode, a data capture to disk (in 8K chunks which will not overwrite one another), and a "snapshot" function that saves a screen in memory for later viewing. Lacking are a print spooler and text upload function.

Watford agrees that there still remains much work to do on the program, and expects another release by the end of the year. "I don't plan another release until it is finished and tested," he said. "And there is lots more to come."

According to Watford, future releases of Omega will contain a powerful phone dialing function that will be compatible with any auto-dial modem, and a possible encryption routine for secure, private data transfers. A built-in password-based encoder/decoder would be breaking new ground for TI terminal emulators, and might find use on local bulletin board systems as well as person-to-person data exchange.

The impact of encoded messages on public BBSs has yet to be assessed. "I think it's obnoxious," said J. Peter Hoddie, Sysop of the Boston Computer Society's TI BBS. "The Sysop wouldn't be able to monitor such traffic...there is so much time, effort, and money invested in a BBS and so little appreciation already." Hoddie also voiced concerns about criminal liability of a Sysop for providing the means of exchange for illegal messages that were encoded. In the past, courts have ruled that "private" messages on a BBS were still accessible to a Sysop and hence the Sysop could not claim ignorance of illegal dealings. However, since the encryption protocol Watford proposes would clearly preclude Sysops from viewing encoded messages, it is possible that ignorance of illegal dealings could be properly claimed.

The central issue has been argued many times, not the least of which being in the program protection/piracy debate. Should users be given power that can be abused? Just as a track copier enables the copying of protected software and could be instrumental in illegally distributing copyrighted software, one of message encryption's uses could be to transfer illegal information over public networks without the knowledge of the network provider. There are legitimate uses to each as well, though; for example, users could justifiably want extra security in their sensitive communications with one another.

For those who will be obtaining the current version of Omega, some known bugs follow. "On rare occasions the scroll routine freaks," said Watford. "A clear-screen character fixes it, as does pressing <FCTN>L to clear the screen...[I] looked for the problem for ages, but could not make it repeat...I finally gave up." Watford added that the display routines are being rewritten for the final version, which should eliminate the bug.

RLE mode does not ignore carriage return-linefeed combinations. The

problem has already been found and corrected in the development version, but the released version must be used with a system that can suppress carriage returns. CompuServe and many other systems can and do suppress carriage returns in ASCII RLE transfers (Omega must receive RLE data in ASCII form). The <FCTN>B buffer and programmable keys fail to send an 80th character, if there is one. A space for the 80th character should avoid that problem. The program also provides a status line across the top of the screen. The line displays baud rate, RS232 port, and buffer size, and can be optionally hidden from display. However, it does not always accurately reflect saved defaults. The defaults are in effect—the status line may display the wrong information for port or baud rate, though. Also, the log dump does not always send the pause character to prevent the other system

from continuing to send data while the dump is in progress.

Another planned enhancement to Omega is full scripting, allowing users to compose on-line session activity before the event and then allow the computer to run unattended. Eventually, says Watford, the user will be able to have the computer dial through a series of numbers, gathering pre-selected data from each one, including file transfers or message reading/leaving. For users of pay networks this could be especially convenient, since messages could be both read and written offline and then quickly transferred while on-line, saving connect time costs.

But even without all these projected enhancements, Omega Version 1.0 is a program to try. It responds to a CHARA1 character set present on disk, so switching to it from another program is rather painless...and the features that it does have are worth the effort of obtaining it.

Besides Omega, Travis has other powerful programs in production. "I have been working on some assembly routines for BBS programs," said Watford. "Output routines will allow variable widths; input routines will

continued on page 348

TIMEX-SINCLAIR

Sinclair Survival Column

by Mark L. Fendrick

Last month we had a look at a few programs from Novelssoft of Toronto, Canada. As promised, this month we examine their remaining offerings for the T/S 2068 (and Spectrum) computer.

One program which I have heard quite a bit about from many of you readers is Timemachine, a BASIC compiler for the T/S 2068 and Spectrum computers. (As we discussed last month, all of Novelssoft's software comes with a T/S 2068 program on one side of the cassette and the Spectrum version on the other.) Timemachine handles not only the integer operations that many earlier Sinclair compilers could, but also can handle floating point arithmetic. This compiler can handle programs that are up to 27K (30K for the Spectrum) without the use of tape/microdrive swapping procedures.

Using Timemachine can be broken down into just seven steps which start with 1) LOAD Timemachine; 2) LOADING (or typing) the program to be compiled; 3) Check to see that your program does not contain any of the commands and/or functions which are not supported by the compiler; 4) Insert a special REMARK line at the beginning of your program; 5) RUN your program and check for possible errors; 6) Compile the program; and 7) SAVE your compiled program. (As with the programs which we discussed last month, there is a sample program supplied which you can compile, giving you the opportunity to work

with a program which you know to be bug free.)

Before you compile your BASIC program, there are a number of compiler directives which you may place in your program. These are a means of giving instructions to the compiler and is in the form REM ! followed by a Sinclair keyword.

The only compiler directive which is not optional is REM ! OPEN #, which turns the compilation on. This enables an entry from BASIC at the point of this directive. To return to BASIC from a compiled program you would use the REM ! CLOSE # directive.

Other compiler directives include:

REM ! LEN—tells the compiler how much memory to reserve for non-dimensioned string variables. For example, REM ! LEN D\$ < - 5 would allow the string defined as D\$ to be up to 5 characters long. This is a memory saving device, for without this directive, non-dimensioned variables would be assigned a length of up to 255 characters. Adding a REM ! LEN \$ < - 32 will limit all variables to 32 characters.

REM ! USR—allows you to specify where you want your compiled code to start. The default location is the top of memory as specified by the system variable RAMTOP. If you use this directive, you can then SAVE your compiled code, LOAD it back into the proper memory location, and then execute it from that address.

REM ! INT—designates the named

continued on page 352

TI Forum continued from page 182

allow messages to be input as one continuous string, without <enter> [required after each line]. [Also,] two screens, one forty column that shows what the caller sees, and a second 32 column screen under the Sysop's control to display status, etc. The Sysop will be able to switch back and forth at will.

"The assembly portion will handle the message base directly, allowing the size of a message to be limited only by the disk space and freeing up a good deal of BASIC programming space. There will also be a routine that searches a file for a string," he said.

Much like Omega, such a routine would introduce some new features to the user that had not previously been seen in the TI world. And the BBS routine will also probably end up public domain, said Watford. "I'm programming for fun, and maybe recognition," Watford declared. Besides, he added, there is probably little money to be made through a commercial effort. "Others have [almost] convinced me that public domain will eventually hurt the TI community. [Hence,] future offerings may be done differently."

Watford was apparently referring to the argument that the presence of public domain programs of commercial quality will run commercial pro-

grammers and distributors out of the market—and with them, other programs that do not have public domain counterparts.

Charlton Also Hard At Work

Although BBS and pay network usage may be down, it seems that telecommunications utilities continue to appear. Paul Charlton, Myarc programmer and developer of the fairware Fast-Term terminal emulator (some would say that Fast-Term is the standard by which all other terminal emulators are judged in the TI world), is hard at work on a project that has interesting possibilities.

Charlton describes his newest program as "...a com-

mand language for Fast-Term which will let you write programs and do anything from the program which you can normally do from the keyboard...you could write a BBS in this language." Stay tuned for more details of this as they become available. Charlton cautioned that the project is far from done.

9640 Speculation Continues

The 9640 review is on the way—as soon as the 9640 arrives, the reviewing can begin. Recent operating system problems have slowed the release of the Myarc machine (also affectionally called the Geneve), but the OS is rumored to be near its first completion (Version 1.0), including full

TI-99/4A emulation. Since the operating system is disk-based, updates can be easily released to registered users of the computer.

"On CompuServe's TI Forum, a heated exchange continues as to the impact the 9640 can have on the TI world and the computing world in general. Overall, the mood seems to be one of tolerance and expectation. "If they are able to get the thing off the ground and have a decent marketing plan," said forum member Jeff Bunting, "then it just might take off... Only time will tell."

Both Bunting and Cynthia Becker, a 99'er from Seattle,

continued on page 350

Applying The Atari continued from page 347

misspell a word you get a wrong answer. This program, like your number to English one, works with groups of three numbers starting at the low position.

I also try to use structured programming, starting at the top and working downward. I use GOSUBs to leave the routine where at and then come back to the same place. And I try to use GOTOS only to move downward within the

same routine. I also use GOTO to go back to the top to do another problem.

K. Dwaine Williams
Odin, Illinois

Many thanks for sharing your program. I have modified the program so that it will work on an 8-bit Atari. String handling is the major difference between 8-bit Atari BASIC and GWBASIC. Subscripted strings do not exist in Atari BASIC, so these had to be simulated with one big string (Q\$). Also, lines containing MID\$, RIGHT\$ or other

string functions had to be modified to perform the proper string manipulation in Atari BASIC. A comparison of the differences between Atari BASIC and Microsoft BASIC with a discussion on how to convert one to the other would be a good topic for a future column.

Using Program Perfect, type in the English to Number Converter and RUN it. You will be asked to enter a written number. Type in the number and press Return. The computer will respond with the

numerical equivalent. Be sure to spell out all numbers correctly, and to appropriately add hyphens or the word "AND." See the following examples:

You Type: Two Hundred Sixty-Three Thousand
Computer: 163,000.00
You Type: Four Million Two Hundred Thousand Three Hundred Fifty-Eight and 10/100
Computer: 4,200,358.10

Next Month

More reader mail, Atari news and 8-bit programs.

Readers' questions, comments and contributions are welcome. Please enclose a self-addressed, stamped envelope (SASE) for a personal reply.

"Program Perfect" is a utility used to check for typing errors when entering programs from this column. Readers can send \$5 for a diskette or a SASE for a listing of this program.

Address all correspondence to: Jeff Brenner, "Applying The Atari 10/87", c/o Computer Shopper, P.O. Box F, Titusville, FL 32781-9990. ●

ENGLISH TO NUMBER CONVERTER

```
DTJ 10 REM ENGLISH TO NUMBER CONVERTER
WPJ 20 REM COPYRIGHT 1987 DWAIN WILLIAMS, USED BY PERMISSION
RSJ 30 REM ATARI VERSION BY JEFF BRENNER
NJJ 40 OPEN #1,4,0,"K:"
PVJ 50 DIM Q$(320),A$(100),R$(32),ANS$(100),CMP$(100),TMP$(100),TMP2$(100)
DGJ 60 DIM NUM1$(100),CHK$(100),CHK1$(100)
MTJ 70 Q$(1)=CHR$(32):Q$(320)=CHR$(32):Q$(2)=0$
NPJ 80 RESTORE 110:FOR I=0 TO 31:READ A$:Q$(I+1),I+10+LEN(A$)=A$
APJ 90 R$(I+1)=CHR$(LEN(A$))
KNJ 100 NEXT I
NJJ 110 DATA ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,ELEVEN,TWELVE
OFJ 120 DATA THIRTEEN,FOURTEEN,FIFTEEN,SIXTEEN,SEVENTEEN,EIGHTEEN,NINETEEN
TDJ 130 DATA TWENTY,THIRTY,FORTY,FIFTY,SIXTY,SEVENTY,EIGHTY,NINETY,HUNDRED
MNJ 140 DATA THOUSAND,MILLION,BILLION,TRILLION
DPJ 150 PRINT "ENTER WRITTEN NUMBER TO CONVERT:"
KNJ 160 INPUT #1;A$
WBJ 170 L=LEN(A$):FS=0:FD=0:NUM1$=""
FDJ 180 FOR X=L TO 1 STEP -1
ILJ 190 IF A$(X,X)="/" THEN FS=X:GOTO 220
HOJ 200 IF A$(X,X)="-" THEN FD=X:GOTO 230
VEJ 210 NEXT X
GNJ 220 IF FS>0 THEN GOSUB 900:GOTO 260
GGJ 230 IF FD>0 THEN GOSUB 970:GOTO 260
YHJ 240 IF X=0 AND FS=0 AND FD=0 THEN NUM1$=".00"
TNJ 250 IF X=0 THEN X=L
TOJ 260 ANS$="":CMP$="":CNT=1:DIG=0
FRJ 270 FOR Z=X TO 1 STEP -1
GLJ 280 IF A$(Z,Z)=CHR$(32) THEN GOSUB 390:GOTO 310
TDJ 290 IF A$(Z,Z)="-" THEN CMP$="Y"
OJJ 300 TMP$=ANS$:ANS$=A$(Z,Z):ANS$(LEN(ANS$)+1)=TMP$
YHJ 310 NEXT Z
EJJ 320 GOSUB 390
KJJ 330 PRINT :PRINT NUM1$:PRINT :PRINT
GJJ 340 PRINT "ENTER ANOTHER NUMBER (Y/N)? ":GET #1,N:PRINT :PRINT "
SUJ 350 IF N=89 THEN 130
VGJ 360 IF N=79 THEN END
QDJ 370 GOTO 340
PWJ 380 END
MPJ 390 LL=LEN(ANS$)
EJJ 400 CHK$=ANS$
JRJ 410 IF CHK$(LEN(CHK$))=CHR$(32) THEN CHK$=CHK$(1,LEN(CHK$)-1)
ELJ 420 ANS$=CHK$
SNJ 430 IF CMP$="Y" THEN GOSUB 450:ANS$="":RETURN
RKJ 440 GOSUB 560:ANS$="":RETURN
DTJ 450 FOR V=1 TO LL
MQJ 460 IF ANS$(V,V)="-" THEN 490
YKJ 470 NEXT V
BCJ 480 RETURN
WFJ 490 CHK$=ANS$(V+1,LL)
TXJ 500 CHK1$=ANS$(1,V-1)
```

```
EJJ 510 GOSUB 560
FXJ 520 CHK$=CHK1$
ELJ 530 GOSUB 560
MMJ 540 CMP$=""
BAJ 550 RETURN
EHJ 560 FOR K=1 TO 32:KK=K-1
LHJ 570 IF Q$(KK*10+1,KK*10+ASC(R$(KK+1)))=CHK$ THEN 600
YBJ 580 NEXT K
DEJ 590 RETURN
FFJ 600 IF K>27 THEN GOSUB 740:RETURN
UBJ 610 N=K:IF N>19 THEN GOSUB 710
IVJ 620 IF N>9 THEN CNT=CNT+1:DIG=DIG+1
FLJ 630 CNT=CNT+1:DIG=DIG+1
RHJ 640 TMP$=NUM1$:TMP2$=STR$(N)
HBJ 650 IF N<10 THEN NUM1$=TMP2$(1,1):NUM1$(LEN(NUM1$)+1)=TMP$
FRJ 660 IF N>9 THEN NUM1$=TMP2$(1,2):NUM1$(LEN(NUM1$)+1)=TMP$
TIJ 670 IF Z<=1 THEN 700
IFJ 680 TMP$=NUM1$
JYJ 690 IF DIG/3=INT(DIG/3) THEN NUM1$=",":NUM1$(LEN(NUM1$)+1)=TMP$:CNT=CNT+1
AXJ 700 RETURN
GKJ 710 IF CMP$="Y" THEN N=N-18:RETURN
EJJ 720 N=(N-18)*10
BAJ 730 RETURN
GVJ 740 VA=0:IF CHK$="AND" THEN RETURN
QMJ 750 IF CHK$="HUNDRED" THEN VA=2:VA=VA+DIG-(CNT-1)
SVJ 760 IF CHK$="THOUSAND" THEN VA=3
XVJ 770 IF CHK$="MILLION" THEN VA=6
XQJ 780 IF CHK$="BILLION" THEN VA=9
EJJ 790 IF CHK$="TRILLION" THEN VA=12
DRJ 800 IF VA-DIG=0 THEN RETURN
LJJ 810 DIFF=VA-DIG
TRJ 820 FOR T=1 TO DIFF
LVJ 830 TMP$=NUM1$:NUM1$="0":NUM1$(2)=TMP$:DIG=DIG+1
GDJ 840 CNT=CNT+1
UYJ 850 IF DIG=0 AND DIG/3=INT(DIG/3) THEN 870
CDJ 860 GOTO 880
DMJ 870 TMP$=NUM1$:NUM1$=",":NUM1$(LEN(NUM1$)+1)=TMP$:CNT=CNT+1
YNJ 880 NEXT T
BHJ 890 RETURN
JGJ 900 FOR Y=X-1 TO 1 STEP -1
TFJ 910 IF A$(Y,Y)=CHR$(32) THEN 940
YHJ 920 TMP$=NUM1$:NUM1$=A$(Y,Y):NUM1$(LEN(NUM1$)+1)=TMP$
YOJ 930 NEXT Y
NFJ 940 TMP$=NUM1$:NUM1$=",":NUM1$(LEN(NUM1$)+1)=TMP$
UMJ 950 X=Y-1
BFJ 960 RETURN
CAJ 970 NUM1$=""
RHJ 980 FOR Y=X+1 TO L
TCJ 990 NUM1$(LEN(NUM1$)+1)=A$(Y,Y)
AZJ 1000 NEXT Y
XTJ 1010 X=FD-1
DFZ 1020 RETURN
```

TI Forum continued from page 348

feel that the 9640 could appeal to markets other than the TI one. "[Myarc] has to approach more than just the TI community with this fantastic computer," Becker said. "[They have] to package advertising and marketing in such a way as to extoll the virtues of the machine to the PC community...no mention of the 99/4A need/should be said to these people...they can find that out for themselves!"

Warren Agee, a talented programmer from Detroit, sees Myarc's scope as a bit more limited, at least for the beginning. "Ain't no way Myarc is

gonna do battle with Big Blue IBM and survive. No way. It took Apple four years to make even an inch of headway in the corporate market, and only because of...desktop publishing." Agee suggests aiming the 9640 at "vertical" and "niche" markets and "not the corporate arena."

The discussion continues. In the coming months, this space will contain the latest news and views on the 9640 machine. That's all for now.

Ron's Part...

Nice guy, great program! I received a nice letter from Donn Granros, author of Old Dark Caves II and, now, "Legends." He wanted to let

me know the beta-test copy of Legends I briefly mentioned in the August column was just a taste of what the final program will be. He and co-author Ed Johnson have spent over a year producing this marvelous software. Donn described some of the work for us:

"We spent an enormous amount of time on Legends. Most of the assembly language code was written from scratch and Legends can pull off a few stunts that I don't believe have been done on the TI before.

My personal favorite is a routine that stores all variables in low memory expansion before switching program sections. This is much faster than storing and restoring from files

on disk. It also does another interesting thing. Stop the program with FCTN 4 (Clear) and type "RUN" and press Enter. The program will get all variables from low memory and will restart with no problems. Nothing lost.

I feel that for Ed and myself this was an opportunity to show that the limits of our TIs have not yet been reached. If in its own way Legends helps keep users from migrating away from the TI then we will be very proud indeed."

Like I said, a nice guy. The program should be available now from Asgard Software (P.O. Box 10306, Rockville, MD 20850; 301-559-2429). It will require 32K memory ex-

pansion and at least one disk drive. Cost: \$27.95. Worth it. Highly recommended.

More from Asgard...

When Warren Agee first wrote "Recipe Writer," a super-fast, dedicated database for keeping recipes, Asgard promised to develop a series of support disks for the database with recipes already keyed in and accessible from "Writer." Well, not only has that come true, they have produced an upgrade to Recipe Writer, itself. Version 2.0 of Recipe Writer is ready to go (\$19.95). Also, check out the "Electronic Gourmet" series of cookbooks

continued on page 351

SOFTWARE

PROBLEM: Bad F5Call error code returns.

MODULE: Clock

Sombody left the '#' sign off of a LDB #E\$illegal Argument

Offset	Old	New
0191	D6	C6

PROBLEM: Non-efficient use of screen memory.

MODULE: GrDrv

Opening a 40 column screen should use the last 2K of an 8K screen block if it's free for use. However, apparently a bad Def was used in MW's source code and GrDrv cannot match an internal code as a 40 column screen.

Offset	Old	New
033A	84	86

PROBLEM: Cannot have more than one IRQ device at a time.

MODULE: IOMan

While Clock gets the size of the IRQ table from the Init module (as it should), IOMan has a different size hard-coded in. Clock inserts the first entry at the front of the IRQ table, but the next call starts searching at the end of the table...which turns out to usually be the header of the first module in your bootfile. Symptoms: If your disk drive is still going (waiting for motor time-out), you cannot Iniz a ModPak device. Or, if you Iniz a ModPak device, your drives will never shut off.

IOMan really should be fixed, but easiest patch is to the INIT Module, to change the number of IRQ/IRQ devices down from 15 to say, 12.

Offset	Old	New
000C	0F	0C

PROBLEM: SS.Montr GetStt Possibly bad.

MODULE: CC30

Although the manual doesn't mention it, CC30 also supports getting the current monitor type set by MonType. The value (0,1,2) is returned in the X register. The code in CC30 should have been a STD RSX instead of STB RSX though.

Offset	Old	New
07D2	E7	ED

PROBLEM: SS.ScTyp returns wrong value.

MODULE: Grfnt

This GetStt call should return a screen type of 1 for 40 column text screens. Instead, it will wrongly return a type of 2. Internal comparison made using wrong register.

Offset	Old	New
0A5B	81	C1

PROBLEM: CLEAR to End of Screen code ignored.

MODULE: GrDrv

L-II windows have a bug where the Clear to End of Screen code doesn't work on the next to last line. They do some calcs to see (after they've done a Clear-to-End-of-Line) if there's a need to clear more lines. The bug is that they used the wrong comparison operator, a BLE (branch if less or equal).

Offset	Old	New
1417	2F	2B

MANUAL

SUBJECT: Creating GFX Windows

Section: BASIC09 Reference

PAGE: 9-37

Here they tell you how to create a graphics window, but show the "merge systdfonts >hw1" AFTER the wcreate. Nope. All you get is dels on the screen. You must merge systdfonts BEFORE opening any gtx windows, unless you care to do a FONT command to that window after merging. They had it correctly on the page before (9-35) about merging so that you can type later.

SUBJECT: F5FORK, F5LINK, F5LOAD, ISCREATE, ISMAKDIR, ISOPEN

SECTION: OS9 Tech Reference

PAGE: 8-16, 8-23, 8-26, 8-49, 8-56, 8-58

On all of these, after the call X should be pointing to the \$BD (carriage return) at the end of the string.

SUBJECT: F5FORK

SECTION: OS9 Tech Reference

PAGE: 8-15

The Y register contains the parameter area size in BYTES, not in pages.

SUBJECT: F5TIME

SECTION: OS9 Tech Reference

PAGE: 8-40

To be exact, on exit X points to the time packet returned to the area at (X) that you had originally passed for the call.

SUBJECT: ISDELETE

SECTION: OS9 Tech Reference

PAGE: 8-50

On return, X should be pointing to the beginning of "MEMO."

SUBJECT: FSALARM

SECTION: OS9 Tech Reference

PAGE: 8-66

FSAlarm is a user call, too. And they left out how to use it.

This call has several variations, which have to do with setting time variables that the Clock module will try to match once a second. You may clear the alarm setting, read it, or set it for one of two exclusive actions. It depends upon the A:B (D) register as to what action FSAlarm takes:

D = 0000 : clear the alarm setting CLEAR

X = ptr to 5-byte time packet (YYMMDDHHMM) SET BELL ALARM

D = 0001 : cause the CC30 "beep" for 16 seconds after the time packet sent matches system time.

X = ptr to spot for time packet return IN-USE CHECK

D = 0002

X < current alarm setting packet returned at (X)

D < current proc id and signal pending in A:B SET SIGNAL ALARM

X = ptr to 5-byte time packet (YYMMDDHHMM)

A = proc id to signal on time match

B = signal to send on time match

SUBJECT: F5DATLOG

SECTION: OS9 Tech Reference

PAGE: 8-78

Actually, not a bad example, but only if you're running on a machine with 4K blocks. On the CoCo 3, Output X = \$4329. The actual code just multiplies B*\$2000 and adds it to X.

SUBJECT: SS.RDY

SECTION: OS9 Tech Reference

PAGE: 8-113

On devices that support it, the B register will return the number of characters that are ready to be read. Both CC30 and ACIAPAK support this feature.

SUBJECT: SS.MOUSE

SECTION: OS9 Tech Reference

PAGE: 8-125 on

Sombody forgot the two reserved bytes between Pt.ToTm and Pt.TTt. As printed, offsets after ToTm are off by two. So insert a "rmb 2 - reserved" after Pt.ToTm. Ignore the system use note at the end after Pt.Sz.

SUBJECT: SS.DSCRN

SECTION: OS9 Tech Reference

PAGE: 8-143

Also, if you specify screen number zero (Y=0000), then you will return to the normal VDG (32x16) screen. This should be done before a SS.F5Scr if you wish to return to a text screen.

A New Columnist

continued from page 188

when the program is "packed" down to I-code (intermediate-code) and executed by RunB, the BASIC09 run time module that is included on disk. Additional BASIC09 subroutines like SysCall (for system calls), Inkey (polls the keyboard), and GFX2 (a graphics interface module) enhance the use of BASIC with Level 2.

It would be unfair for me not to mention that some folks are having problems with disk operations, especially with older controllers. Because Level 2 OS9 operates in the CoCo 3's 2MHz, high speed mode, some of these controllers are reporting errors of all sorts (if you can boot up into the operating system at all)! Keep this in mind if you are considering a purchase.

All in all, I like Level 2 OS9! It is a vast improvement over the Level 1 version, and with the windowing, I am hopeful that with Tandy's support, that it will take off. For the price, this system can't be surpassed! The Level 2 OS9 operating system for the CoCo 3 costs \$79.95 and is available at your local Radio Shack store.

Inside Level 2 OS9

An interesting book *Inside Level 2 OS9*, written by Kevin Darling, is currently the definitive source for information for the serious Level 2 OS9 programmer. Containing an in-depth look at the OS9 Level 2 shell, device/file/window descriptors, and how the memory management unit works, the book is a Level 2 hackers dream. Additionally, source codes and explanations for some new modules that keep track of paths, windows, the module directory are included with the book.

The book sells for \$39.95 and is available from Frank Hogg Laboratory, 770 James Street, Syracuse, New York, 13203.

TI Forum
continued from page 350

on disk for "Writer." Volume #1, a "Salute to the Southwest," contains spicy recipes for drinks, appetizers, and dinners from Texas. Volume #2, "Appetizers," is a large collection of great meal-starters containing everything from meatballs to dips to vegetable creations. Future columns covering meat entrees, poultry entrees, side dishes and desserts. Finally, I can get Mrs. Albright interested in using the computer! Best of all, the disks are only \$6.95 per volume and, of course, require Writer 2.0. Neat idea, especially with the traditional cheaper-than-cheap Asgard pricing. Chris Bobbitt, president of Asgard, continues to support the 99/4A with both barrels, so let's keep supporting him.

More Than 80 Columns With TI Writer...

Here is a quick tip from Tom Kennedy of Seattle, Washington. It relates to folks who may want to print 132 (in condensed mode) with TI Writer. Tom explains:

"There are two steps to over-80 column text in TI-Writer:

1) Get to the first line of your text. Now type the following keystrokes:

CONTROL-U SHIFT-O(not zero) CONTROL-U

The C-U enters/exits the "Special Character Mode," which allows you to insert HEXidecimal numbers into your text. The SHIFT-O generates a tiny "OF" (HEX-OF = DEC-15 = Condensed on an Epson or Star Gemini). The zero in "OF" will actually/appear as a dot above a tiny little "F" on the screen. That sets the printer, now to format the document.

2) Where needed in your document (usually at the top) insert a blank line with the following sequence:

.RM 132;FI;AD <Carriage Return>

The line must start at column zero, with a "." for the first character, and a <CR> as the last. You could also include a "LM xx" in there to set a left margin if you want, and of course you can change the "132" to whatever column you want. Insert a ".NF;NA" when you want to go back to the original text width. For more info, see page 146 of the T-W manual about "Special Character Mode," and the Formatter section for Format commands."

This month's winner...

We are very pleased with your enthusiastic response to our software give away. Your response serves two purposes:

first, it allows us to put some of the software generously given to us by producers for review to good use and, two, provides input to the editor of *Computer Shopper* that our column is being read. And that—reader mail—is what determines if the column survives in print. So, please! Keep sending in those postcards to enter the drawing. We will continue to give away stuff as long as you show us you want it.

This month's winner is a

lucky soul for sure! John W. Bullard of Lafayette, LA wins a complete set of the terrific monthly, "Super 99 Monthly." These collector's items (the newsletter was renamed "The Smart Programmer" in April of 1986) are all the issues of Super 99 from September, 1984, to March, 1986. They are packed with lots of programming information that should be of interest to Mr. Bullard. Also, we are including the great graphics program "Joypaint 99," the "Joypaint Pal" and

"Clip Art #2" companion disks, all from Great Lakes Software (804 E. Grand River Avenue, Howell, MI 48843). Mr. Bullard, you have hit the "mother load!" Congratulations! Join in the fun, folks, drop us a postcard with your address on it, and you can be eligible to win as well. All it costs is 14 cents to "roll the dice!"

John Calvin Traver...

One of my dearest TI friends, Barry Traver (editor of

the marvelously successful "Genial Traveler Diskazine") reports that his son, John Calvin, has taken the robes of the entrepreneur himself. JCT is offering a set of 6 floppy ("flippy" means, in essence, each side of the disk is a SS/SD disk, so 6 floppies = 12 SS/SD disks!) disks containing John Calvin's selections of the best public domain and freeware software available for the TI. And he should know—he has

continued on page 352

UPS SYSTEMS HEADQUARTERS **KNAPCO** **RELIABLE COMPUTER BACK UP POWER.**

When Your Operation Depends On A Computer, Make Certain Your Computer Can Depend On Its Power. Rely On Knapco UPS Systems . . .

Electric Power—vital, indispensable—yet a source of frequent unpredictable problems

Blackouts

- Total loss of commercial ac power
- Result from causes beyond utility's control
- Such interruptions often cause equipment damage and/or loss of processing data and memory if equipment is not protected

Brownouts

- Planned power reductions by utilities because of excessive demand
- When the voltage falls under minimum limits, self-protection devices will cause a shutdown

Electrical Noise (static, interference)

- Airborne power line pollution induced by electromagnetic radiation or electrostatic coupling
- Caused by radio/TV broadcast transmitters, poor grounding, lightning, heavy equipment and utility network switching
- Results in spurious errors, erroneous operation, slow degradation of equipment components

Voltage Transients (spikes, surges)

- Momentary high-frequency spikes on a 60 Hz voltage wave form
- Caused by brush-type motors in office equipment, improper grounding, lightning, etc.
- Results in erratic equipment operation via spurious command signals, register of valid signals, memory loss, program damage, data errors, damage to components

Voltage Instability (sags, flickers)

- Surges and high-level increases when large loads are suddenly disconnected
- Flickers occur when the power company switches substations because of lightning or other problems
- Caused by large loads such as air conditioners, elevators, etc., coming on-line
- All of these can result in sluggish or erroneous operation, failure to operate, memory losses, damage to drive motors

Frequency Deviation


- Causes improper operation of computer timing and logic circuits

ANSI Standards

- Provides power sig. at your instance
- Allows additional drop due to interconnecting impedances

Quantity Pricing Available
UPS Systems Add \$15.00 For Shipping

Knapco UPS Systems



FACTORY DIRECT
LOWEST PRICES ON UPS SYSTEMS IN USA

UNINTERRUPTIBLE POWER SOURCE

IT WORKS

UPSC 200 + 200 WATT \$279.
RETAIL \$499.

UPSC 350 350 WATT \$299.
RETAIL \$799.

UPSC 550 550 WATT \$399.
RETAIL \$899.


UPSC 800 800 WATT \$599.
RETAIL \$999.

UPSC 1000 WATT \$699.
RETAIL \$1250.

★ EUROPEAN DESIGNED UPS

Made In USA Modified Sine Wave
2 Ms. Switchover Time
All Models w Internal Geli Celis
Unconditional 1 Year Warranty
4 Outlets, Brown Out, Black Out
RFI And Spike Protection


DATA TRANSFER SWITCH



AB SERIAL \$29.50
ABCD SERIAL \$44.50
AB PARALLEL \$34.50
ABCD PARALLEL \$44.50
AB CROSSOVER \$42.50


MODEM SPIKE PROTECTORS
MSP-1
\$19.95

LINEGUARD 6 OUTLET BLOCK



12 @ \$9.95
24 @ \$17.95 3 MOY.


ISO LINE STRIPS



6 OUTLET BAR
W/6 FT. CORD 12 @
BAR 6-1 \$19.95 \$14.50

12" MONITOR
Hi Res. Electrohome/Canada
\$69.95 10 @ \$50.

8" DISK DRIVE




PARADISE
SERIES FDD 100-00
SERIAL DRIVE UNIT
SUPPORT SUPPLY
FAN PLUS A BEAUTIFUL
ENCLOSURE, CONTROL PANEL,
WITH CABLE. PARADISE NEW
NOT USED NOW \$9.95

AUTOMATIC VOLTAGE REGULATORS

INPUT RANGE 85-135V
EMI-RFI + SURGE
Wattage

AVR 500 \$ 95.00
AVR 1000 \$159.00
TVR 2000 \$339.00
TVR 3000 \$495.00



AMERICAN POWER CONVERSION UPS Systems Made In USA

	Retail	Dealer
300 PC+	\$ 459—	\$299.
330XT+	\$ 499—	\$309.
450 AT+	\$ 799—	\$499.

Synchronized Sinewave Models

800 RT	\$1099—	\$699.
1200 VX	\$1399—	\$905.

2 To 3 Ms. Transfer Time Compact Design
Prevents Surges, Sags, Spikes, Noise and Interference

add 10 for shipping and handling (non-shipping charge 2.50) FAX 813 3924504

call us at **(813) 392-0406**

H.J. Knapp of Florida, Inc.
4750 96th St. N. St. Petersburg, Florida 33708

Sinclair Survival continued from page 182

variables which would follow the directive as being the INTEGER type. These variables may consist of single character variables, named variables and/or arrays.

REM ! INT + —is the same as above, but identifies the named variables as being of the POSINT type.

REM ! INT FN—tells time-machine that certain named functions will return a value in the restricted range of -32768 thru 32767.

REM ! INT + FN—serves the same purpose as the last directive, however the restricted range is positive (0 - 65535).

REM ! FN (INT) & REM ! FN (INT+)—define certain variables as being INTEGER or POSINT when used as dummy variables in DEF FN statements.

REM ! LIST—causes the program to produce a listing of the runtime routines used and the machine code variables. Addresses are shown in both decimal and hexadecimal.

REM ! LINE—will print the address of the code which represents a given line in the uncompiled BASIC program.

REM ! PRINT—switches the output for both the LINE & LIST directives from the screen to the printer.

When you are writing your BASIC program you will have to be aware of the fact that certain commands take on a slightly different way of operating, while some will not exist at all. An example of this would be the automatic error trapping when using the INPUT keyword. Asking for input requires that the proper type of data is entered or else the program will simply repeat the input request. The use of a STOP command will cause the compiled code to stop executing and control to return to the BASIC operating system. The DIM statement requires a little thought since TimeMachine handles it slightly different than in Sinclair BASIC: Only 2 dimensions are allowed as opposed to the virtually unlimited number allowed in BASIC (you are only limited by the amount of memory you have available.) The dimension you declare are permanent, however, and you cannot reDIMENSION later in the program. TimeMachine does support both numeric and string variables.

TimeMachine does not sup-

port the following Sinclair BASIC keywords; CLEAR, CONTINUE, ERASE, FORMAT, LIST, LLIST, LOAD, MERGE, MOVE, NEW, RESET, SAVE, VERIFY or VAL\$. You can employ these commands in your programs by returning to BASIC, issue the command and then return to your compiled code if necessary.

You may also use other machine code routines along with your TimeMachine compiled code as long as the two (or more) routines do not overlap in their location in memory. Long programs may be compiled in sections if necessary, but with some losses in efficiency. There is a section in the manual which deals with this.

During the actual compilation process, there will be a number of things for you to look for on-screen. If something occurs during the compilation of your program which does not fit the rules set forth by the compiler, you will receive one (or more) error messages indicating what the error is. Appendix A of the user manual describes each of these errors, possible causes and where to look for more infor-

mation concerning the problem.

Once all of the preliminary preparation of your BASIC program is complete you will now attempt actual compilation of the program. Typing *C will start the compilation process which is a three pass procedure. The first pass, which is indicated by a magenta border, checks for unsupported commands and recognizes all of the compiler directives you have included as well as DIM statements. The second pass, identified by a cyan border, is a chance for the compiler to determine how long the final compiled code will be, and determine the destination addresses of the GO TO and GO SUB statements. The final pass, recognized by the white border, generates the actual machine code. Because the compiler uses the video display to store information (as most compilers do), you will see this information in the form of screen garbage which is no cause to be alarmed.

Those of you who have read this column for any length of time know that I have always warned you to approach any compiler with extreme caution as they did not handle enough to make them useful. This was

my standard warning until I encountered TimeMachine. Earlier compilers did not handle such important factors as floating point arithmetic or string variable arrays. Without these (and other non-supported functions) no truly useful program could be written and compiled. Cameron Hayne, the author of TimeMachine, has answered these problems and as a result I can recommend this program for not only its ease of use, but its usefulness. Both Mr. Hayne and Novelsoft are to be commended for bringing TimeMachine to the market.

The final offering from Novelsoft is modestly called The Worx!. Written by partners Ariel Frailich and David Ridge, it is the most comprehensive organizer program available for the T/S 2068 (and of course, Spectrum) computer. Like all organizer software for the other computers, The Worx! contains a number of functions to make life easier, or at least more organized! They include a planner to track day to day appointments, a notepad to file reminders, messages lists etc., a telephone listing for up to 50 entries, a calculator, a clock/timer/time zone feature, a biorhythm chart producer, a decision maker which will help you weigh up to 20 factors against each other, and a system I/O routine.

The Worx! comes set up for use by either cassette or microdrive operation, but can be customized for any other configuration you might have. When you load the program you are asked for the current date. After the date is checked and confirmed, you are asked for the number of days in the current month, and if the year is a leap year or not. (This could be improved by including a routine which would automatically know the number of days in a given month, or if the year entered is a leap year. The formulas for this exist and would speed up the initialization procedure.)

The Planner section of the program starts off by displaying the calendar of the current month along with the menu of options. Choosing Activities loads all of the entries with today's date into the work file and then sorts them by time. You are allowed a maximum of 20 entries for any given date. When you choose the Add sub-function you will first be prompted for a time using a 24 hour clock. Once the time has been designated you will have an opportunity to enter a memo of up to 17 characters long. The Delete sub-function will allow you to remove a

TI Forum continued from page 351

his father's 1500 disk collection to choose from. The ensemble includes the "best of the best"—from utilities to full-blown application software. Well worth having. Send \$24

to one locally, check there for the set to J.C. Traver, 835 Green Valley Drive, Philadelphia, PA 19128.

In The Public Service...

The U.S. Department of Health has made available a pamphlet of the Surgeon

General's report on "AIDS" (or Acquired Immune Deficiency Syndrome). As a physician, I have seen the increasing toll this horrid disease is taking and face the concern of other patients about their "catching" the disease. The Surgeon General's report provides current information on AIDS and will calm many fears of the American public if they would read the facts about the illness. What the heck has this to do with a TI computer column? Well, just this. Jack Shattuck of the Delaware Valley Users Group (DVUG) has done his part in making this information available to all. He typed the entire pamphlet into TI Writer, word for word (1600 plus lines of text). Put them on disk, and mailed them to 20 or so of the larger user groups across the country for distribution to their memberships and to anyone else requesting the information. A monumental undertaking by Jack and certainly shows that one person can help make a difference. You can get the disk by sending a self-addressed disk mailer and return postage to the Central Westchester 99'ers Users Group, 1261 Williams Drive, Shrub Oak, NY 10588. Art Byers, a prime mover in that fine group, has volunteered to help distribute the disk in this manner for Jack. It should be available in many other users groups as well, so if you belong before sending off to Art. To

Jack and Art: its folks with your compassionate and giving spirit that make so many things possible. With public education, this disease may move away from panic and closer to understanding.

From The Mailbag...

My hats off to James W. Cox, treasurer, of the M.U.N.C.H. ("Massachusetts Users of the Ninety-nine and Computer Hobbyists," whew!) users group. James forwarded an almost complete set of that groups newsletters. An amazing recapitulation of the lives and times of the 99ers—how they have come to grow with their machines as they have understood them more. M.U.N.C.H. is really one of the pioneering TI groups, having been in existence now since September of 1981. They are well-known throughout the TI user base as one of the very best. You can contact that group at 560 Lincoln Street, P.O. Box 7193, Worcester, MA 01605. Also, a new newsletter appeared from a group I was familiar with by the grapevine route. The "Johnson Space Center Group" (J.U.G., 2321 Coryell Street, League, TX 77573) sent along their latest issue. It confirms the excellence of the group which is in the grapevine. Now is the time to join a user group. Now, more than ever you need the support and news a group can offer. ●

continued on page 354

COMPUTER SHOPPER

CUSTOMER SERVICE

WE'RE DEDICATED TO OUR VALUED CUSTOMERS!

CALL

1-305-269-3211

(Mon.-Fri. 8 a.m. - 5 p.m. EST)

1. Change of address.
 - Please notify us 4 weeks in advance.
2. Any questions about your subscription.
 - Your subscription expiration date.
 - Delivery difficulties.
 - Missed issue.

3. Renew your subscription early.
 - Guarantee uninterrupted delivery of **COMPUTER SHOPPER**.

OR WRITE US:
COMPUTER SHOPPER
Subscription Service Dept.
P.O. Box 1419
Titusville, FL 32781

(Please include your label or a facsimile with all correspondence.)