Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 13 Number 6 November/December 1996 $6

A report on the Chicago TI Faire — Page 10

Barry Harmsen, left, and Bruce Harrison are all smiles (Photos by Gary Cox)

Geneve 9640

MDOS utility routines and how to write new ones — Page 16

The Art of Assembly

Source code that gives you everything you ever wanted in full bit-map mode — Page 14

Getting more out of TI-Writer

Editor and formatter tricks you may never have heard of - Page 32

Beery Miller sold CD-ROMs containing TI programs.

Bruce Harrison named Birdwell award winner

Reviews

Speed Reading

P-GRAM Reloader

Assembly Poker

AMS Video Titler

Casino (Klondike) Solitaire

Getting started with c99 (a new series) — Page 6

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published bimonthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Periodical postage paid at Round Rock, Texas. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

Display advertising deadlines and rates are available on request.

All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680. We cannot take responsibility for unsolicited manuscripts but will give consideration to anything sent to the above address. Manuscripts will be returned only if a self-addressed stamped envelope is included.

Foreign subscriptions are $40.25 (Mexico); $42.50 (Canada); $40.00, surface mail to other countries; $52 airmail to other countries.

All editions of MICROpendium are mailed from the Round Rock (Texas) Post Office.

Mailing address: P.O. Box 1343, Round Rock, TX 78680.

Telephone & FAX: (512) 255-1512

Delphi TI NET: MICROpendium

Internet E-mail: jkoloen@io.com

John Koloen .....Publisher

Laura Burns .....Editor

*READ THIS

Here are some tips to help you when entering programs from MICROpendium:

1. Most BASIC and Extended BASIC programs are run through Checksum, which places the numbers that follow exclamation points at the end of each program line. Do not enter these numbers or exclamation points. Checksum is available on disk from MICROpendium for $4.

2. Long Extended BASIC lines are entered by inputting until the screen stops accepting characters, pressing Enter, pressing FCTN REDO, cursoring to the end of the line and continuing input.

FROM TEX COMP LTD.

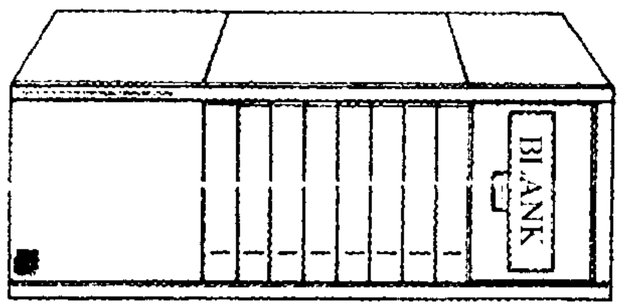
WE AT TEX COMP WOULD LIKE TO THANK MICRO-PENDIUM READERS FOR THEIR SUPPORT AND WISH ALL OF YOU A HAPPY HOLIDAY SEASON.

PARSEC ATTACK INVADERS TOMB STONE

SUPER BUY!

HOLIDAY SPECIAL!
SALE PRICE $35

ADD $20 FOR S/H PER P-BOX



P-BOXES NEW / RECONDITIONED UNITS IN ORIGINAL TI PACKAGING WITH INTERFACE CARD, POWER CABLE, AND P-BOX INSTRUCTION MANUAL FROM TI.

ANY OF THESE 4 POPULAR TITLES **$ .99 ea.**

STOCKING STUFFERS FOR ALL AGES

DATABIOTICS CARTRIDGES ARE GOING FAST THIS MAY BE THE LAST CHRISTMAS THESE TITLES ARE AVAILABLE FOR THE TI.

$6.00

MYSTERY GRAB BOX (INCLUDES 5 GAME CARTS)

THIS BOX MIGHT HAVE ANY OF THE PRODUCTS WE SELL, BUT WE DONT KNOW EXACTLY WHAT IS IN EACH ONE. WE DO KNOW THAT THEY HAVE AT LEAST 5 GAME CARTS IN EACH, PLUS OTHER GOODIES FOR YOUR TI.

OTHER CARTRIDGES ON SALE

TE II	$.99
PROTECTOR 2	$.99
TAX INV/REC KEEP	$.99
HUSTLE	$6.95

SANTAS PICK FOR CHRISTMAS

THE VERSITILE AND CONVENIENT

MAGIC MEMORY $49.95

MAGIC MEMORY belongs in every TI99/4A, give your TI the gift of power. Magic Memory brings many of the average users needs right into one handy cartridge. With features like disk manager functions that rival DM1000, and editor assembler loader functions. It also offers the advanced user easy access to many programs that are used regularly like memory manager, advanced sector editor, debugger, and an editor/ assembler cartridge that does not need the editor or assembler files off the disk (a real time saver) all in one convenient cartridge. MAGIC MEMORY comes with a 58 page owners manual that will help you get started and explain all the features contained in the cartridge.

DBT DataBioTics

4/A FLYER	$24.95
BEYOND PARSEC	$24.95
BEYOND WORD WRITER	$39.95
BLACK HOLE	$24.95
BOXER	$24.95
BREAKTHGH/ BREAKOUT	$24.95
BURGER BUILDER	$24.95
CONCOLE CALC	$24.95
D-STATION I	$24.95
D-STATION II	$24.95
DESKTOP PUBLISHER	$39.95
ESCAPE	$24.95
FACE CHASE	$24.95
MAGIC MEMORY	$49.95
MIDNIGHT MASON	$24.95
MINI WRITER	$24.95
MUNCHMAN II	$24.95
PARAPRINT	$69.95
TI PLANNER	$24.95
PROTYPER	$24.95
RED BARON	$39.95
SPACE PATROL	$24.95
SPY'S DEMISE	$24.95
STARGAZER	$39.95
STAR TRAP	$24.95
TI TOAD	$24.95
TI WORKSHOP	$49.95

SOLID STATE CARTRIDGES

ORDER BY PHONE

VISA & MASTER CARD EXCEPTED

TEX COMP LTD.
AMERICAS NUMBER ONE TI COMPUTER RETAILER
425 E. ARROW HWY. #732
GLENDORA, CA 91740

(800) 846-3474 FAX (818) 339-1134 **ADD 5% SHIPPING {$5.00 MINIMUM}**

ALL PRICES ARE FOR CASH/CHECK, ADD 4.5% FOR CREDIT CARD ORDERS

COMMENTS

Congratulations to Bruce Harrison

Regrettably, I was unable to attend this year's Chicago TI faire. And it sounds from reading the report by Gary Cox that a good time was had by all.

He was most enthralled by hardware developed in Europe by Michael Becker. His GPL card boosts the speed of the TI by eight-fold! That's a remarkable improvement. A U.S. distributor is being sought for Becker's products. I hope one is found soon.

Another piece of news from the faire is that Bruce Harrison was named the recipient of the John Birdwell Award for service to the TI community. Bruce is well-deserving of this recognition. Over the years he has developed so much software for the TI that it's almost impossible to own a TI or Geneve and not have at least one of his programs. Bruce is one of those guys who is always willing to help out someone with a software problem. He is prolific not only in the volume of materials he's produced, but in his willingness to go the extra mile to help other TIers. Congratulations, Bruce!

ALL'S NOT WELL ON THE NET

Once you've gotten used to e-mail, you wonder how you ever got along without it. This comes to mind because in mid-November my Internet provider upgraded its e-mail server and I and thousands of other subscribers ended up without e-mail for about five days. This struck me as incredible. Five days!

What really bothered me about it was that the technical staff kept saying it would be fixed "in another two hours." This was the standard response I got when I inquired as to when I could retrieve my e-mail. This wouldn't have been as much of a hardship as it turned out to be if we didn't rely on e-mail to move documents between our Geneve and the Macintosh that we use to layout MICROpendium pages. Five days thus became an eternity. Finally, the e-mail started operating properly but then I had to leave town for a business trip that lasted four days, thus putting us way behind on our schedule for MICROpendium.

I was so angry that in addition e-mailing the provider a nastygram (they could receive e-mail, they just couldn't upload it to subscribers), I called a large, national Internet provider to find out details about changing over to their system. I dialed their technical support line and, wouldn't you know, I got a voicemail that said "Our e-mail system is temporarily out of service. Our technicians are currently working to fix the problem. Please call back later."

Needless to say, I didn't.

MICROPENDIUM OFF GENIE

MICROpendium is no longer subscribing to GEnie. We'd been subscriber ever since GEnie started, but the rate structure it adopted this year finally drove us offline. That and the huge reduction in support for the TI made the decision inevitable, but not easy. We're still available on Delphi and, of course, the Internet.

HAPPY HOLIDAYS

Laura and I hope everyone has a safe and happy holiday season. We'll see you next year!

—JK

BUGS & BYTES

Small world

This item was posted on the Internet TI newsgroup (comp.sys.ti) by Jon Kruse.

Just thought I'd share this with you.

I recently received some TI programming books from Eunice Spooner. In this package, I obtained a book that I had originally owned a long time ago. Not just a copy of the book though, I recieved the SAME book I had gotten rid of about 8-9 years ago. I recognized my handwriting and doodling immediately. Small world, eh?

Living dead

When it comes to computers, some of us prefer to call them "orphans," but science fiction novelist Bruce Sterling has a list server including them (along with panoramas and the Inca quipu) as "dead media." His Dead Media Project has information on dead media findings. To join his forum, send him a message at bruces@well.com.

COMPETITION COMPUTER

800-471-1600 (Nationwide & Canada ORDERS ONLY)

350 MARCELLA WAY MILLBRAE, CALIFORNIA 94030

MasterCard VISA

TI-99/4A™ CD ROM

CD ROM UPDATE: CAN NOW BE USED WITH PC99 STAGE 3A USING NEW PC99 UTILITY DLCONV.EXE! 5 ISSUE SUBSCRIPTIONS $100 SINGLE (120 MB) ISSUES NOW AVAILABLE $35.00

WINTER READING:
SERVICE MANUAL TI MONITOR(PANASONIC) $20
TI WAFERTAPE USERS MANUAL $8
TI 99/4A HEXBUS INTERFACE PERIPHERAL SOFTWARE SPEC $5
TI THERMAL PRINTER THEORY OF OPERATION & TROUBLESHOOTING GUIDE $3.50
TI HOME COMPUTER SYSTEM MEMORY CRU & INTERRUPT MAPPING SPEC
& ADDITIONAL MEMORY MAPPING INFO $6
P CODE CARD THEORY OF OPERATION $8.50
32K MEMORY CARD THEORY OF OPERATION & TROUBLESHOOTING GUIDE $12
SPEECH SYNTHESIZER THEORY OF OPERATION $3
TI EXTENDED BASIC ASSEMBLY LANGUAGE CODE PRGRAMMERS GUIDE $5
PERIPHERAL EXPANSION SYSTEM THEORY OF OP & TECH TRAINING MANUAL $15
TI DISK CONTROLLER TROUBLESHOOTING GUIDE $8
TI 99/4 SYSTEM SOFTWARE DESIGN SPECIFICATION $4
GPL PROGRAMMING LANGUAGE PROGRAMMER'S GUIDE $25
HEX BUS INTERFACE PERIPHERAL MANUAL $5
STAND ALONE DISK MEMORY SYSTEM THEORY OF OPERATION $4
ORIGINAL VERSION OF DISK MEMORY SYSTEM OPERATING INSTRUCTIONS $8
TI TERMINAL EMULATOR PROTOCOL MANUAL $7.50
TI 99/4A HOME COMPUTER SERVICING MANUAL 3RD REVISION 10/83 $15
TI GPIB ADAPTOR CARD MANUAL(IEEE488 CARD) $5
TI VIDEO CONTROLLER OPERATING MANUAL $6

OTHER GOODIES:
UMATIC TAPES TO USE WITH VIDEO CONTROLLER $5
NAVARONE HOROSCOPE MAKER(DISK) $12
ASGARD TRIS-2 (WITH MUTANT BLOCKS)CARTRIDGE $20
SUPER MARIO BROS EXBASIC DISK $5
CC40 NEW IN BOX $80
0007 SUPER AGENT(K+Q) $5
SKETCHMATE(SAVE SUPERSKETCH PICS ON DISK) $10
STAR PATROL (K+Q)$5
GOLF TOURNAMENT(K+Q) $5
CONTROL DATA PLATO SOFTWARE:
NOT RELEASED BY TI
15206000 BASIC NUMBER FACTS(SPEEDWAY GAMES)$5
15206030 WHOLE NUMBERS(PINBALL GAMES)$5
15206020 DECIMALS(DART GAMES) $5
15206010 FRACTIONS (DART GAMES) $5
15206070 COMPUTER LITERACY:INTRODUCTION(90PAGE MANUAL)$12
15206080 PHYSICS: ELEMENTARY MECHANICS(GRADE 12)$5
READY TO RUN PLATO AVAILABLE $3/DISK
MANY ORIGINAL CONTROL DATA CLAMSHELL PACKAGES AVAILABLE

415-696-1108

COMPETITION COMPUTER
350 MARCELLA WAY
MILLBRAE CA 94030
ORDERS:1-800-471-1600
INFO:415-697-1108
FAX(CALL FIRST)415-697-7406
WE ACCEPT MASTERCARD,VISA,AMERICAN EXPRESS,AND DISCOVER CARDS
PERSONAL CHECKS(7 DAYS TO CLEAR) OR MONEY ORDERS
WE SHIP PRIORITY MAIL(2-3DAYS TO US ADDRESSES)
NO HANDLING CHARGES NO SRCHARGE FOR CREDIT CARDS
DID YOUR CHILDREN OUTGROW YOUR TI EDUCATIONAL OR OTHER SOFTWARE?
TRADE IT IN FOR OTHER TI SOFTWARE AND RECEIVE CREDIT EQUAL TO 50%
OF OUR CURRENT RETAIL TOWARDS PURCHASE OF OUR TI SOFTWARE
OUR 2/96 CATALOG HAS 800 ITEMS FOR YOUR TI AND WE HAVE ADDED 100'S

FEEDBACK

Where is TI ramp to info highway?

Thanks for the article about the TI list server in the September/October issue. I e-mailed my subscribe request about a week or so ago. Last night I was on America On-Line, checked my e-mail, and lo and behold, I had more TI-related messages and postings than I had time to read in one evening! It was strange, but great, to see famous names from the TI world that I have known for years through the pages of MICROpendium attached to e-mail messages on *my* computer! Of course, the messages weren't *to me*, but there they were, nevertheless. It felt like "old home week"!

Maybe I've missed something again, but I have read several articles lately in MICROpendium that seem to imply that TIers are surfing the Internet, and maybe even getting information from the Web. Surely they're not doing this on a TI! I do venture out on AOL, but have trouble accessing/finding anything on the Internet, and AOL flat denies me access to the Web. The messages say that *neither* of my DOS machines — an 8086 AT&T 6300 and an 8088 XT, both with 640K and 32- and 20-meg hard drives, respectively — have the horsepower to browse the WWW. If I can't with my hardware, how can someone with a TI or Geneve do it?

Other than Internet access, after three years with a DOS machine, the only things that impress me about them are the same things that forced me off the TI in the first place — 80 columns, a hard drive and a real keyboard. If my DOS computer didn't have those three advantages, I'd be back on my TI — it's vastly superior in many ways. For example, GW-BASIC is a primitive joke, compared to TI Extended BASIC. And nobody seems to know anything about assembly on IBM compatibles. I'm tired of hearing about C+++, Virtual BASIC and all the other whiz-bang languages that the PC magazines carry on about, but no one in the real world actually knows anything about.

It's good to see that people are still not only selling TI software and hardware, but actually developing new hardware for it! (By the way, it's nice that we now have a SCSI HD controller for the TI. However, I don't know of anyone who knows of anyone who has ever *seen* a SCSI drive offered for sale, new or used, let alone seen one in person!) I'd love to have a hard drive for my TI — *and* an 80-column screen *and* a real keyboard. I am negotiating now to purchase a 386 in a few days. If I do the deal, the *three* IBM-compatible machines together will not have cost me what it would cost for my three wish-list items above for a TI!

Okay, enough whining. Keep up the good work.

Arnie Stewart
Edgewater, Florida

As you surmised, no web browser exists for the TI, though quixotic speculations have been floated about developing one. You can get text-only access to the Internet and WWW, but not through AOL — Delphi, and, to a limited extent, CompuServe, provide it, as well as local providers in some areas. — Ed.

One of the best

Extended BASIC debugging, TI99 assembly, Geneve assembly, FTP sites, a TI list server, AMS card memory — this may be one of the best MICROpendiums ever (September/October 1996). Thanks.

John C. Johnson
Cedar Rapids, Iowa

Speed Read needs no AMS card to run

Some customers were confused by the box on page 22 of the September/October issue. Because the Speed Read and AMS Titler products were together, and the headline was misleading, they assumed that Speed Read was for use with the AMS Card. It's *not*. Speed Read requires only disk drive, 32K memory and either Editor/Assembler or Extended BASIC. Please put some kind of clarification in the next issue, so that potential Speed Read customers won't be "scared off." Thanks.

In recent weeks, I've been working in collaboration with Mickey Cendrowski on an improved version of her Load Master (fairware) Program. With some help from assembly language, we've been able to put together a much improved product. Among other things, the new version allows running of E/A Option 5 and E/A Option 3 programs directly from the catalog listing.

I'll be introducing a new product for the AMS card at the Chicago Faire Nov. 9. My "testers" report that it works perfectly. The AMS card has proven to be a real wonder, and I expect to be making more products for it in the near future.

Bruce Harrison
Hyattsville, Maryland

You say goodbye, but I say hello

Thank you for thinking of me. I had to give up the teaching/setting people up aspect because the doctor determined it was too much.

However, I am unrestricted in my personal use of the TI and continue to use it for service here doing mail lists (I have 101 mail lists) and providing the Sisters and letter writers with labels, etc.; the shaky hands here are so grateful.

I have not given up the TI, and I didn't want my friends to think I have. Mine is probably one of the busiest ones in the TI community! I simply had to redirect my energies to a different type of service. Mail lists are the biggest now, but I still do banners and cards, and when I finish my family history will be very much back into graphics.

It sounded so final under a farewell in Bugs & Bytes (September/October 1996). I am very much TI and with the TI it's always "Hello," not "Goodbye" from me!

Sister Pat Taylor
Dubuque, Iowa

Sister Pat lives up to the adage, "Bloom where you're planted" as a resident of a nursing home. — Ed.

Send your letters and comments to *MICROpendium Feedback*, P.O. Box 1343, Round Rock, TX 78680.

Beginning c99 — Part 1

Getting started

By VERN JENSEN

Let me start by introducing myself. I'm a long time TI owner and programmer, and many of you know me as the author of the Extended Basic games Maze Mania and The Castle, as well as the c99 game Virus Attack. (Each of these games has been reviewed in the Micro-Reviews column.) I originally bought my TI back in 1987 as a "Game Machine," since I had heard that some games could be purchased for as low as $3 each. I then discovered TI-BASIC, and have been a programming addict ever since.

Since I enjoy playing games, I naturally wanted to make games as well. I was thrilled when I discovered that with Extended BASIC you can add sprites and other advanced features to your program. However, when creating Maze Mania and The Castle, I quickly ran into the limits of Extended BASIC, the main one being speed. I spent much effort making The Castle run as fast as possible, and even added some assembly routines for special effects, but the game was still much slower than I wanted. Also, since it ran as slowly as it did, I couldn't add many features that I wanted to add to the game, such as enemies and moving platforms.

Naturally, I wanted a way to create games that avoided these problems; games that ran fast and could handle the features I wanted to implement. However, assembly language was not the answer, as least not for me. I had already tried learning it, and it was simply too complicated. I wanted something that was easy to use, but powerful. Could something like this possibly exist? And then I discovered it — c99, which is truly easy to use and also very powerful. However, I still had a problem: I didn't know C, and all the books that taught C were made for the IBM or Macintosh.

Since I had no way of learning how to use c99, I gave up on it for a while. Then after a while our family purchased a color Macintosh, and I also picked up a C compiler for it, which enabled me to learn how to use C on the Mac. Once I had learned C, it was easy to learn how to use c99, and I started programming my first game using c99, Virus Attack. If you don't believe that c99 is powerful, or you don't believe that it is fast, I suggest that you take a look at Virus Attack. It will change your mind if anything will! (Virus Attack is a shareware game similar to Dr. Mario that can be ordered from me for $15.)

STARTER KIT

Now that I know how to use c99, I'd like to share my knowledge with you, so that you too can become a c99 programmer without having to go through what I went through, and without having to buy a Macintosh. To make this process easier, I have assembled a five-disk DSSD "c99 Starter Kit" that gives you everything you need to use c99 all in one easy-to-use package. The disks are organized, each disk having its own category:

1) The c99 compiler, set up to load from Funnelweb.

2) Documentation.

3) Libraries.

4) Example programs.

5) Some great c99 Libraries made by Bruce Harrison.

Even if you already have the c99 compiler, I would strongly recommend that you order this disk set, since it is well organized, and contains the new c99 compiler version 5.0, rewritten to go hand-in-hand with FunnelWeb, that also compiles faster than older versions. Also, this series will assume that you have these disks, since I can then give you precise step-by-step instructions on how to use the c99 compiler, giving you exact information on where you can find each file and so on. You can order this disk set from me by sending $5 (to cover S&H) to the address at the end of this article.

To use c99, I recommend that you have at least two DSSD disk drives, and having a RAMdisk or hard drive would help, although they certainly aren't required. I also highly recommend a printer, as it can be useful to print out your source code to help you track down bugs. Although it is possible to run c99 from only a single SSSD disk drive, I wouldn't recommend it, since you would have to constantly swap disks to go back and forth between the editor, compiler and assembler, as well as having separate disks for your source and compiled code. Also, my five-disk "c99 Starter Kit" won't work on a SSSD drive, so you'd have to order a copy of c99 from someone else, such as a mail order dealer. I myself have two DSSD drives, and get along just fine, although a RAMdisk or hard drive would be nice, since it would greatly speed up the compiling and assembling process.

GETTING STARTED

The rest of this article is written assuming you have ordered and received the "c99 Starter Kit" mentioned above. I suggest that you order this kit, and continue reading this article once it arrives. I will also assume that you have two DSSD disk drives. If you don't, then it will be up to you to determine when you need to swap disks. This issue you will learn, step by step, how to type in a simple c99 program, compile it and run it. Next installment we will start to compare c99 with Extended BASIC and talk about the differences between c99 code and Extended BASIC code, so you can start to learn how to write your own programs in C.

To get started, insert the c99 disk called "c99 Libraries" in drive 1 and a blank disk in drive 2. Then use a Disk Manager to copy the files CSUP, GRF1, and GRF1;H from drive 1 to drive 2. Now remove the c99 Libraries disk from drive 1 and insert in its place the disk named "c99 Compiler." Leave your other disk in drive 2. Disk 1 is your "work disk" and contains the editor you will use to write your source code, the c99 compiler which compiles it into assembly language, the assembler which translates the assembly language code into machine code, and the Editor/Assembler loader, which lets you run your completed program. All of these programs are easily accessible from FunnelWeb. Disk 2 is where you will store your source code and the libraries needed by your program. Unlike Extended BASIC, which has everything you need all ready to go, c99 requires you to load "libraries" that contain the functions you want your program to be able to use. All c99 programs need the CSUP library you

(See Page 8)

c99 —

(Continued from Page 7)

copied a moment ago from the Libraries disk, and the GRF1 library contains many functions for displaying graphics and sprites that are very similar to the familiar Extended BASIC functions that perform the same task. We'll talk about libraries more in later issues.

Now that you have the libraries you need, boot up FunnelWeb (disk drive 1) using either the Extended BASIC or the Editor/Assembler cartridge (to boot FunnelWeb using Editor/Assembler, use option 5 — "Run Program File" and type "DSK1.FW") If you loaded from Extended BASIC, press 2 ("Edit/Assm") from the main menu to get to FunnelWeb's Editor/Assembler menu. If you loaded from the Editor/Assembler cartridge, this menu will automatically appear. You can toggle between Funnel-web's Editor/Assembler menu and the TI-Writer menu by pressing the space bar. You know you are looking at the Editor/Assembler menu if the second menu option says "Assembler."

Once you get to FunnelWeb's Editor/Assembler menu, press option 1, "Program Ed," to load the program editor. This editor is similar to TI-Writer, but it doesn't save information about the layout of the document, as this data could confuse the C compiler. When the editor has finished loading, type E and press enter to start editing the document. Then type in the source code for the c99 program listed after this article. Don't worry about typing in the wrong number of spaces between words, since c99 is flexible and doesn't care how many spaces you leave between words. This gives you the freedom to adopt any coding style you wish. Also take note that there are both uppercase characters and lowercase characters in c99, unlike assembly language and Extended BASIC.

EDITOR COMMANDS

In case you're not familiar with the Program Editor, here is a list of some commands you can use:

FCTN-0: Toggle line numbers on/off
FCTN-1: Delete character
FCTN-2: Insert character(s)
FCTN-3: Erase line
FCTN-4: Move down a page
FCTN-5: Move right half a page
FCTN-6: Move up a page
FCTN-7: Tab
FCTN-8: Insert line
FCTN-9: Return to menu at top of screen
FCTN-E,S,D,X: Move cursor

> The compiler only "sees" the first six letters of each variable and function name, and will only report the six letter version when reporting an error.

Once you're ready to save the file, press FCTN-9 to return to the menu, then type SF (SaveFile) and press enter. When prompted for the filename, type "DSK2.TEST;C" and press enter. Once the file is saved, press FCTN-9 again to return to the menu, type Q, and press enter to quit. When asked if you really want to quit, type E (Exit), press enter, and you should be returned to the Editor/Assembler menu. Now it's time to compile your program! Select option 4, "C-Compiler," to load the c99 Compiler. Once it loads, you will be prompted for a response for several options. Press enter twice to select the default of "No" for the first two options, but type Y for the last one ("Assume Long Jump?". There is no need to press enter after typing Y. A future installment will explain these first three options, but for right now we just want to compile the program.

When prompted for the Input Filename, the name of the last file you worked with — "DSK2.TEST;C" — should come up. This is what we want for the input file, so just press enter. Next you will be prompted for an output filename. Modify the text to read "DSK2.TEST;S"; the "C" in the first filename means that the file is C source code, and the "S" in the second means the file is assembly language source code. After you finish typing in the output file name, press enter, and the file will be compiled. Since it is a very small program, it will compile quickly. If there are any errors the C compiler will report them, in which case you should write them down and then open the file back up with the editor and correct any typos you have. I should also mention that the compiler only "sees" the first six letters of each variable and function name, and will only report the six letter version when reporting an error. So, for instance, if an error were reported on a line that has the variable "keyCode" the compiler will only display "keyCod"; don't let this confuse you. When the compiler is finished, type N (No) when prompted if you want to rerun it.

ASSEMBLING THE PROGRAM

Once back at the Editor/Assembler menu, select option 2, "Assembler," to assemble the program. Once it loads, all of the fields should be filled out for you. The source file name should already read "DSK2.TEST;S" and the object file name should read as "DSK2.TEST;O"; you don't need to worry about the "List device name" field; just leave it blank. Press enter three times to get down to the options field, and press FCTN-1 once to delete the "R"; this should leave you with only a C. The R stands for Registers, which c99 code doesn't use. Removing the R should make it assemble slightly faster. The C stands for Compact, which will make your code smaller, so we will leave that option on. Press enter again, and then FCTN-6 to proceed. Or, if you made a mistake, you can press FCTN-8 to change the fields. When the assembler is finished, press enter to return to the Editor/Assembler menu. (It should take about as long to assemble as it does to compile.)

Now it's time to run the program! Type 3 — "Loaders" — to get to the E/A Loaders menu. Then type 4 for "Load and Run"; you should then see "DSK2.TEST;O" on the screen. Press enter, and the object code for your program will be loaded. Now you need to load the c99 support library and the graph-

(See Page 9)

c99 —

(Continued from Page 8)

ics library. Press FCTN-3, type "DSK2.CSUP" and press enter, then press FCTN-3 again, type "DSK2.GRF1" and press enter. Next, clear the text by pressing FCTN-3 once more and then enter, leaving the line blank. Now you should be presented with a screen containing all the names you can use to try to start the program. Move to the name "START" by using the arrow keys (FCTN and E-S-D-X) and press FCTN-6 (Proceed). If you accidentally hit enter instead, just hit it a few more times to get back to the first screen. If everything goes well, you should see the screen turn purple and then a blast of characters being displayed on the screen. (Pretty fast, isn't it?) Congratulations, you've just run your first c99 program! (Don't worry, the next program will be a little more interesting.) When you're done watching it, press FCTN-plus to reset your computer.

Next time, you'll start to learn how to make your own programs in C. I will show you a more interesting c99 program and an Extended BASIC program that does the same thing, and discuss what is different about writing the C code. This will teach you the differences between C and BASIC, which will enable you to quickly understand how to write C code. I'll also explain what some of those compiler options do, tell you an easier way to load your programs, and describe how to create c99 programs that can be loaded from E/A Option-5.

If you have any questions or comments, I'd love to hear them. Just send them to Vern L. Jensen, 910 Linda Vista Ave., Pasadena, CA 91103, or you can email me at Vern_Jensen@lamg.com.

GRF1

```
#include "DSK2.GRF1;H"

main()
{
int chr, keyCode, status;

Grf1();
Screen(14);

chr=33;

do
{
HChar(1,1,chr,768);
chr++;

if (chr >= 126)
chr=33;

keyCode = Key(0,&status);

} while (status == 0);
}
```

Beware of fraud in cyberspace

By DAVE HOWELL

This article appeared in the October 1996 issue of On CUE, *the newsletter of the Computer Users of Erie. —Ed.*

It is estimated that Internet users lose $100 million per year to fraud. Many of the schemes are aimed at older people — of whom I am one.

Most of what millions of computer users roaming the Internet see is the usually benign electronic pages of health news, entertainment, information and innocent-looking chitchat. But that's not all that lurks behind some of those pages.

There are plenty of computer scam artists who, like telemarketers, push questionable health cures, phony credit-repair scams and grandiose get-rich-quick schemes. And why not? Why should we expect it to be any different in the Internet world? If we want the Internet to remain accessible by everyone, we have to educate ourselves to deal with such human phenomena just as we are learning to cope with telemarketers on the phone.

The Federal Trade Commission recently stopped one such operation — a pyramiding scheme very similar to a chain letter — which is illegal if done through the U.S. Mail. For example, the FTC found that investors, lulled in by a respectable-looking Internet web page, paid anywhere from $250 to $1,750 each. The early investors were paid off with money from later investors — who, in turn, wound up losing their money when the scam operators pulled their web page and vanished.

According to the FTC, about 95 percent of those caught in such scams will lose money. Unfortunately, investors in that scam lost $6 million before the FTC obtained a federal court order stopping the company.

While most web sites are legitimate, a fraudulent site can suddenly appear and just as quickly disappear. The operator stays one jump ahead of the law. This type of fraud via the Internet violates the same federal and state laws as telemarketing fraud, but so far, enforcement agencies have been slow to act. "The Internet is a lot like the Wild West," says Jerry Hancock, director of consumer protection for the Wisconsin attorney general.

One of the reasons slowing the progress of enforcement is due to the fact that the origin of many scams are in foreign countries. Many gambling operations that are illegal in the U.S. are based in countries that allow gambling.

Then there are the Internet hustlers who aren't selling anything — they simply try to get enough information on you to steal as much money as they can. They may send inviting e-mail messages asking for your credit card, bank account or social security numbers. *Anyone dumb enough to send such information uncoded through cyberspace* deserves to lose money — unless, of course, you originated the request.

As Don McLeod, of the *AARP Bulletin* (published by the American Association of Retired Persons), warns, "tricksters, representing themselves as Internet access

(See Page 10)

1996 TI Chicago Faire

New hardware from Europe gains attention

By GARY W. COX

It's hard to believe that 14 years have passed since the first Chicago TI Faire. Many people now in the computer field didn't even know what a computer was 14 years ago. Nonetheless, on Nov. 9, 1996, the 14th annual Chicago TI Faire was held in Evanston, Ill., sponsored by the Chicago TI Users Group.

Hal Shanafield, faire chairman, once again did a great job organizing the event. Estimated attendance was about 100. At times the main floor seemed more crowded than last year. Vendor attendance was about the same but there weren't quite as many items for sale this year as last. At the same time many new products were introduced. And many user groups had a variety of products for sale.

One user group, headed up by Michael Mickelson, was the Windy City TI Users Group. It had the largest array of used software and equipment for sale of anyone at the event. My group, the Mid-South TI99/4A User Group, also had an array of hardware and software for sale, with items such as consoles going for as little as $10 each. Other user groups had a variety of items for sale plus the usual promotions for their groups. Dave Connery manned the Chicago User Group table, offering a variety of software for sale.

Charles Good of the Lima TI Users Group offered for copying the complete Jim Peterson library, at no charge. William Lucid of the Indianapolis TI Users Group had a variety of items for sale as did the Milwaukee TI users group.

Barry Harmsen of the Dutch TI Users Group from The Netherlands, along with Gerd Weissmann of Germany, brought some interesting new TI hardware (mentioned later in this article). The TI Users of Will County sold a variety of items, including flip strips for the TI console.

Someone who had not attended a Chicago TI Faire in several years was Beery Miller of 9640 News. New from Beery is a CD-ROM containing the contents of TI and Geneve public domain and shareware software. Most of the software was obtained from the GEnie software libraries and the 9640 News BBS. Beery was also promoting the new GEnie Interactive service, which is a new Internet-based online service. The new service can be accessed through any Internet provider, linking the user to much of the same information as is accessed through the regular online service.

The most interesting part of the faire was a variety of hardware projects created by Michael Becker and demonstrated by Gerd Weissmann. One such product is the High Speed GPL Card for the TI PEB which allows the TI99/4A to run eight times faster than a base unit!

NEW HARDWARE FROM EUROPE

The most interesting part of the faire was a variety of hardware projects created by Michael Becker and demonstrated by Gerd Weissmann. One such product is the High Speed GPL Card for the TI PEB which allows the TI99/4A to run eight times faster than a base unit! A built on-board EEPROM Programmer allows the storage of up to 16 modules.

Also new was the SGCPU which contains the entire TI99/4A on a card. Right now a small TI99/4A keyboard is needed to operate the device, but soon an AT and option to use an AT keyboard will be added.

Gerd also demonstrated an 80-column card that supports more than 256,000 colors. Michael also has available a DS/DD controller card.

All these products are being made in Germany. Negotiations with a U.S. supplier are under way to allow the products to be ordered from inside the U.S. These are some really great products which, if they become available for sale from a U.S. supplier, should become very popular. Currently these items are available only for order from Germany.

For more information contact: System 99 User Group (SNUG), Michael Becker, Diedesfelderstr.12, 68308 Mannheim, Germany.

Mike Wright of Cadd Electronics was running version 3a of PC99 TI99/4A Emulator for PC's. Work continues on updates, although there is no schedule as to dates on the completion of updates. Mike, indicated the next release, version 4 of PC99, may contain Myarc disk controller support, as well as speech.

Mike is also placing a great many TI manuals on PC diskette complete with all original graphics. An executable program is included with the manuals so nothing additional is needed to view the documents. Users can print out the manuals or do searches for keywords. Most manuals in the PHM module series have already been converted to this online format with more on the way.

SCSI EPROM UPGRADE

Bud Mills, of Bud Mills Services, was present with version 1.07 of the SCSI EPROMs for the TI and Geneve SCSI cards. Bud mentioned that his group is one step closer to completion of all items promised on the SCSI controller. However, currently working with the SCSI controller are SCSI hard drives, Zip drives, and Syquest drives. Testing is under way with JAZ drives.

A new version of the EPROM is being tested and should be available soon. Bud also mentioned he has completed an order of Horizon 4000 boards and thus the Horizon 4000 Ramdisks are once again available.

(See Page 11)

TI FAIRE —

(Continued from Page 10)

Tim Tesch was present at the faire and mentioned he is continuing to work on updating the PORT terminal program and correcting various bugs. Tim is also working on the Geneve LOAD SYS. When completed, and with an updated SCSI EPROM that is in the works, the Geneve will be able to boot from a SCSI hard drive.

Bruce Harrison showed several new programs, including a new version of Loadmaster (version 2.2). Loadmaster is a really neat program which will identify most file types on a diskette and inform the user about is needed to use that file. Thus, if the file it a TI-Artist picture Loadmaster will indicate that fact in the file catalog. If the file is an Extended BASIC or Editor/Assembler option 3 or option 5, it will run the file. Loadmaster will also print a disk catalog in the form of a diskette sleeve. The speed of the program's overall operations is much faster than previous versions. The program is written jointly by Mickey Cendrowski and Bruce Harrison and is distributed as shareware. Loadmaster is available from various user groups, or Charles Good will send anyone a copy by sending $1 to, Charles Good, P.O. Box 647, Venedocia, Ohio, 45894. Be sure to read the title screen for instructions on sending in your donation for the program, it's well worth it.

Bruce also had a new program called AMS Slideshow that uses the AMS card to display TI-Artist pictures. AMS Slideshow can be set to change pictures at a selected interval of up to 0.1 seconds between pictures, or it can be set to look for a keyboard press before going to the next picture. AMS Slideshow requires the AMS memory card to function at it's best. Also new from Bruce was Speed Reader which teaches a person to speed read.

Don Walden of Cecure Electronics was present with a variety of special interest parts, such as replacement GROM connectors, TI99/4A replacement keyboards, Rave 99 Speech Cards, and a variety of CC40 equipment at discounted prices. Don also mentioned he now has a new address and phone number which is listed at the end of this article.

Ken Gilliland of Notung Software talked about plans for additional TI Casino games. Also, Ken's solitaire game is now available as a stand-alone product or as an addition to TI Casino.

RAMCHARGED SELLING ASGARD SOFTWARE

Victor Steerup represented RAMcharged Computer with a large table full of software, including Asgard products not available anywhere else. One item that caught my eye that RAMcharged was selling was Rapidcopy ($9.95), an excellent disk duplicator program. Rapidcopy has been around for a while but it is the best program available for copying diskettes at lightning speed.

New from RAMcharged were several game programs ($7.95 each) for the Geneve, including Train Twister, Time Guardian, Jungle Terror, Submarine Revenge, Sea Teller, Cave Explorer, and Space Champions. RAMcharged also had available a spell checker called Spell-It in SS/SD, SS/SD, DS/DD, and hard drive versions.

BRUCE HARRISON RECEIVES BIRDWELL AWARD

Finally, at the banquet following the faire, Dave Connery received an award from the Chicago TI Users Group for his outstanding work for the group. Then the John Birdwell award for outstanding commitment and contributions to the TI community went to a very deserving person, — Bruce Harrison!

All in all, the 1996 Chicago TI Faire was a success and a good time for all. A videotape was made of the event by the Chicago TI Users Group and should be available soon. I hope to see everyone at Fest West. as well as the Lima TI Faire.

1996 Chicago TI Faire vendor listing

9640 News, P.O. Box 752465, Memphis, TN 38175-2465, (901) 368-1169.

Bud Mills Services, 166 Dartmouth Dr., Toledo OH 43614, (419) 385-5946,

CaDD Electronics, 45 Centerville Drive, Salem, NH 03079-2674, (603) 893-1450, or (603) 895-0119, EMAIL: mjmw@xyvision.com

Cecure Electronics Inc. c/o Don Walden, P.O. Box 132, Muskego, WI 53150; 1-800-959-9640 (orders only), (414) 422-1010 (voice), (414) 422-994A (tech line), BBS (414) 422-9669.

Harrison Software, 5705 40th Place, Hyattsville, MD 20781-1727 (301) 277-3467.

Hoosier TI Users Group, P.O. Box 2222, Indianapolis, IN 46206, email: lucid@indy.net

Mid-South TI99/4a User Group, P.O. Box 38522, Germantown, TN 38183- 0522, (901) 358-0667, Email: garycox@netten.net

Lima TI Users Group, C/O Dr. Charles Good, Ohio State University, Lima Campus, Lima OH 45804.

Notung Software, 7647 McGroarty St. Tujunga, CA 91042, (818) 951-2718 (now carrying MS software), email: empken@loop.com, Web page: http://www.loop.com/~empken/

Ramcharged Computers, P.O. Box 81532, Cleveland, OH 44181, (216) 243-1244

TI-Gruppe Mannheim: Koenigstrasse 17-19 D-67655, Kaiserslautern, Germany.

TI Users of Will County, 1400 Caton Ave., Joliet, IL 60435

Vereniging TI-Gebruikersgroep (Dutch TI Users Group) le Osterparkstr. 141e 1091 GZ Asterdam, The Netherlands.

Windy City 99 Club, 1549 Webster Ln., Rosemont, IL 60018-1423, (312) 444-5680

Back to BASIC(s)

The joy of programming goes hand-in-hand with the joy of learning

By DICK BEERY

This is reprinted from the Spirit of '99, *the newsletter of C.O.N.N.I.—Ed.*

It's great to have friends. One of mine, Ken Marshall, receives a lot of computer magazines, and when he is finished with them he passes them along to me. In the most recent batch I found the January 1996 issue of *Family Computing*, targeted at users of the Macintosh, PC, and multimedia. Starting on page 55 there is an article written as, I believe, part of a monthly column called "Double Click." The columnist is a lady named Robin Raskin, and her thesis is that, for reasons you will see in a couple of paragraphs, we need to return to the teaching of computer programming to children and others.

It needs to be stated at the outset that in one sense, Robin and I will be talking apples and oranges. Only partially. True, she is talking about the computers and platforms already listed. I am talking to users of the TI99/4A and the Geneve.

My thesis is this: if you are still using and cherishing your TI computer, and I hope you are, I hope you will keep it and use it in the ways I will be suggesting, even if you at some point buy another kind of computer — or have done so already. Most of us are aware that "out there" are an abundance of programs already beautifully written and just waiting for us to enjoy. So why should we write something? Read on.

If you are still using and cherishing your TI computer, and I hope you are, I hope you will keep it and use it in the ways I will be suggesting, even if you at some point buy another kind of computer — or have done so already.

Another difference in perspective between what Robin is talking about and what I will be saying to you is that she is talking about "the newer programming languages [that] are graphical and 'object-oriented.'" That is, she goes on to say, "they provide a set of building blocks for creating programs. Bit by bit, programmers put those basic modules together to create a computing experience that accomplishes what the programmer wants it to" (p. 55). To me, the experience she describes is familiar, as anyone who has worked with either BASIC or Logo will attest.

But, why do any of this in the first place? Robin says "Taught well, programming is absorbing and engaging. A completed program provides a profound sense of mastery and accomplishment."

NOT A PROGRAMMER

Let me speak to that point from my own personal experience. I am not now, and have never been, a programmer. I took a course offered to teachers at my school that included some programming in BASIC for several computers, among them the Apple. My approach was to take a program someone else had started and delete all the things I didn't want and put in the elements I did want. I understand that I am not alone in using this approach to programming.

One of my sons, the one who now programs professionally, said to me one day, "Dad, why don't you write a program of your own, starting from scratch?" My reply was, of course, that I had all the programs I needed, and, if I didn't, someone would probably write just what I needed in the not-too-distant future. He persisted, and finally, to shut him up, I said, OK, I would give it a try.

But what to write? He suggested a LOAD program in TI Extended BASIC, and I reluctantly agreed. After a few false starts, and with some subtle hints from him, I produced a working LOAD program. Nothing fancy. But it worked! It was a real thrill went over me. I would not have felt more proud after having de-

(See Page 13)

FRAUD IN CYBERSPACE—

(Continued from Page 9)

providers or merchants doing business on the Net, make requests such as, 'We need to reconfirm your access code,' 'We want to make sure we have your correct credit card number,' or 'We need your social security number for registration purposes.'"

"You have to understand," says Scott Charney, chief of the Justice Department's Computer Crime Unit, "that when you send a credit card number, if you're sending it clear (uncoded) across the Internet, there are all those different sites where it can be intercepted." No one has yet placed a foolproof security system into operation, although they are working on it.

Even those who know the Internet well can get taken in. They may know so much they will think they're in control until they're in someone else's Web site — then watch out! And it doesn't matter how intelligent one is. There have been cases where rocket scientists, medical doctors and attorneys were victimized.

Be careful!

BACK TO BASIC(s) —

(Continued from Page 12)

signed some architectural wonder such as the Taj Mahal. All because I, yes, that's right, *me*, had created this dumb little program that others could use and benefit from.

Robin goes on to say (and this is, in my belief, the most important thing she says in her article) "Children need to understand that human effort is what makes events happen on a computer screen — and that programmers are mere mortals just likethe end user. Then children can stop thinking of the computer as an infallible deity and know that it is controlled by creative adults (or children) putting together the right instructions.

The process children master in the programming environment can be applied to any sort of problem-solving challenge, and they should be given the opportunity to at least try it." Think about what she just said, "stop thinking about the computer as an infallible deity...." Think how we all, at times, need to apply that principle to other aspects of our lives: the TV commentator or personality whose dictums we often take at face value, without challenging their validity; the costume epic we treasure as "authentic" without searching for further information as to how carefully it was researched; etc.

I read the quoted portions of the paragraph above to another friend, Harley Ryan Jr., and he immediately came up with a beautiful example of total lack of understanding as to what computers are and what they do. It seems he entered a Radio Shack store one day where a young boy was playing on a computer and the mother was browsing shelves of software. Suddenly, the boy called his mother over to the computer. She looked, frowned, punched a few keys and then called the clerk over. He asked what was the matter, and her response was that her son had asked the computer a question that it had not answered. (You know the kind of program I am talking about). The clerk's response was that maybe the answer was not available in the program. The mother challenged that. "What do you mean, not in the program?" and she got even more irate. The man could not convince her that material exists in a program only because some programmer put it there: that humans created and control computers, not the other way around. She grabbed the child by the arm and left in a huff. Extreme? Maybe.

Robin goes on to say, "Educational benefits abound. Programming requires great concentration and care, critical thinking and logic skills. It requires thinking about SEQUENCING [capitals mine] — what must happen first, second, third and so on, for a series of actions to take place. It requires risktaking — trial and error — followed by systematic error correction.

Programmers learn to look at their creations piece by piece and debug problems. They learn to see interconnections between pieces of code and gain an understanding of how pieces can be combined effectively." As a member of the teaching profession, I can tell you I cannot recall having seen the goals and objectives of any school system that did not include all or most of the above.

LEARNING SKILLS

So, how can we use the TI99/4A or the Geneve to help our children develop these critical learning skills? If you yourself program, the job is easy. You already know what to do. But there are pitfalls there. Having a pretty good knowledge of English does not qualify you to teach it to people who speak other languages — or even your own! Part of what goes into teaching, beside endless patience, is learning how to break down into small steps whatever is to be taught and then present it to the learner in some appropriate order. We could go on and on discussing that. But we don't need to. How did my son Brian teach me to program? "OK, Dad, what do you need first? (Get something to show up on the screen.) What next?" etc.

Fortunately, there is help for those who find the foregoing idea too challenging, or who themselves have no experience with programming. Texas Instruments released two cassette-based programs: Teach Yourself BASIC, and Teach Yourself Extended BASIC. They are also available on disks and, I believe, on cartridge. Many newsletters regularly feature how-to programs. Choose the very simplest, basic ones in the beginning. Have your child type in the program as you watch. You can be the "spotter" who notes and comments on typing errors as they occur. Or, you can allow the mistake to happen. Either you will get an error message when you try to run the program, or if the misspelled item is something that appears as letters on the screen while the program is running you can ask, "Gee, should that be spelled that way?" Both techniques allow for the opportunity for debugging — and there are few other ways that meticulous care can be taught as effectively or remembered for as long.

After typing in several such programs, you may ask yourself or the child, "What other kinds of things could a program such as this one do, if one merely changed a few things?" Out of this can come an entirely new program, and the thrill of having created it.

So, if you have convinced yourself of the advantages of having your child (or yourself) learn to program, find some way to make it happen. Ask someone in your users group, if you belong to one. Talk to a friend who writes little programs on his/her computer to satisfy occasional needs. Read the BASIC and Extended BASIC manuals, and type in the models, then try running them. Persist, and you might just experience the "profound sense of mastery and accomplishment," as Robin says it, that a completed program done by you or your child, with or without help, can provide. Write to me, in care of this newsletter, and let me know how this all worked out for you, or ask
questions. Enjoy!

P.S. I recently typed in and debugged a few programs from our C.O.N.N.I. newsletter, and, you know what? I got goose pimples all over again!

THE ART OF ASSEMBLY — PART 61

Full bit-map motion

By BRUCE HARRISON

As promised, we're going all the way this month. We are offering source code that gives you just about everything you could ask for in the Full Bit Map Mode. How about 32 sprites with their own set of 256 character patterns, and automatic sprite motion! In the demo, only four sprites are used, but you can extend them to 32, and everything will still work. You can even use the magnified sprites, either the single character type or the four-character type. For those who get MICROpendium on disk, there's not only the source file as SIDEBAR61, but the object file SIDE61/O, so you can try out this program right away. The program name is the ever-popular START.

Last month, in the Enhanced Graphics mode (AKA Half-Bit Map) we tucked away the Sprite Pattern Table, Sprite Attribute Table, and Sprite Motion table in the lower parts of VDP RAM. In the Full Bit Map case, we still managed to tuck away the Attribute and Motion tables in the space between the Screen Image Table and the Color Table. The Motion Table is at >1B00, and the Attribute Table is at >1B80. The Sprite Pattern Table had to be placed at >3800, and it occupies all the space from there to the end of VDP RAM. We copied the standard character set into that space, then put a special "block" character definition into the patterns for characters 0 and 255

To make this demo, we borrowed some source code from Part 42 of this series, so that a nifty little single-pixel spiral gets drawn on the screen, then the legend "Bit Map with Moving Sprites" is placed at the bottom. Now the main event begins, with four sprites appearing nearly dead-center of the screen, then going into "automatic" motion. As in last month's case, their motion is controlled by the same User Interrupt code. This month's source code includes the PLOT, CHAR and BITSTR routines that first appeared in Part 42. Again, as in last month's program, we've magnified the sprites.

There are some differences in the subroutines we borrowed from Part 42. The SETBM routine, for example, starts by blanking the screen, so that all the setting-up operations are invisible to the user. SETGM, which resets to the Graphics mode, also blanks out the screen, and clears the graphics screen while it's blanked.

The subroutines PLOT, CHAR and BITSTR are unchanged from Part 42, and the subroutines SPRITE and SPRINT are identical to those in last month's sidebar. The SPIRAL routine is also identical to the one in sidebar 42. We've put the data items that are needed by the user interrupt into a section by themselves, just to make that data easier to recognize.

TODAY'S SIDEBAR

Yes, it's another complete but "nonsense" program. As in previous examples, we've stashed away the stuff from the graphics mode, including the color table and pattern table, and those six bytes from the location pointed to by >8370. Next we write the character definitions into the Sprite Pattern Table at >3800, and then supplement those by putting in a solid block pattern for characters 0 and 255.

Now we use SETBM to put us into Bit-Map mode. This is similar in some respects to the SETHB subroutine in last month's sidebar, except that this clears the bit-map screen to all white with a preset black foreground color. We changed this from what was in Part 42 in two ways. First, we added the screen blanking in the first two instructions, so the transition would be invisible. Second, we "folded in" the subroutine CPDT, which used to be a separate one. By the way, if you like different starting colors, change the LI statement just before label CT in the SETBM subroutine to any combination you like.

Upon return from SETBM, our main program writes >E1 to VDP Register 1. That unblanks the screen and also sets up for magnified sprites. We put a copy of that byte >E1 at >83D4, which isn't really necessary in this program, but should be done in your own programs so that a keypress won't destroy what you've done to VDP Register 1. This level of magnification means the sprites use only one character, but they're twice the normal height and width of a screen character. The other levels of magnify will work just as well, but you can try that out for yourself by putting in your own four-character sprite definitions.

Now the program uses the PLOT subroutine in a series of loops to create a single-pixel spiral starting near the upper left corner of the screen and working its way to a spot near the center of the screen. When that finishes, we use the BITSTR subroutine to put the legend "Bit Map with Moving Sprites" at the bottom of the screen in white on green.

THE MAIN EVENT

Now we set our own memory location to allow four sprites in motion, then use the SPRITE subroutine four times to put a red "A," a yellow "B," a green block and a magenta block on the screen as sprites. These start out near the middle of the screen, then move outward in four directions. The B and the magenta block move to the left, while the A and the green block move to the right. The blocks move downward and the letters upward. To start the motion itself, we put the address of our user interrupt at >83C4. Our boy Jean-Guy says this stuff on the screen could hypnotize him. Maybe so. After a while, the four sprites will all converge back at their starting point, so you can observe that their motion is accurate.

Now the program just does a BL @KEY, so nothing happens until you press a key. When you do, the program first clears the word at >83C4 to disable the user interrupt, then uses SETGM to put the VDP back to the normal graphics mode. SETGM blanks the screen first, so you won't see any glitches during the transition. It also includes a loop to clear the graphics screen before it unblanks the screen. Finally, the main program puts back those six bytes at the location pointed to by >8370, so that file operations will work when you're back in the Editor/Assembler's control.

That's it! The subroutines here should prove useful to any reader who's trying to make programs that involve Bit Map mode. If

(See Page 15)

ART OF ASSEMBLY —

(Continued from Page 14)

your program needs to have disk access, you'll probably have to go back to the Graphics mode before doing that, as there really isn't anyplace to do file operations in VDP RAM without overwriting at least part of your sprite descriptor table, even if you've done a CALL FILES(1) in the beginning. Of course there is another way, if you can live with having only 128 sprite character definitions. You could place the Sprite Descriptor Table at >1800, but put actual definitions in starting at >1C00. Your sprite character numbers would then start at number 128 and run through 255. Combining that with a CALL FILES(1) would allow you to perform some file operations even while in bit-map mode. Be warned, though, that your sprites will stop moving while files are read or written, because the DSR does not allow interrupts. For help on doing CALL FILES, see SIDEBAR44, starting at the bottom of page 7 in the February 1995 issue of MICROpendium.

Next issue we'll have another set of routines for you. Among other things, we'll show a quick and easy way to reverse single characters or four-character sprite definitions from left to right and vice versa. See you then.

X-Authentication-Warning: bermuda.io.com: jkoloen owned process doing -bs Date: Tue, 5 Nov 1996 18:00:29 -0600 (CST)
From: jkoloen <jkoloen@io.com>
To: jkoloen@io.com
Subject: sidebar61
MIME-Version: 1.0

SIDEBAR 61

```
0001 * SIDEBAR 61
0002 * A COMPLETE PROGRAM
0003 * DEMONSTRATES FULL BIT-MAP MODE
0004 * WITH AUTOMATIC SPRITE MOTION
0005 * 02 APR 1995
0006 * PUBLIC DOMAIN
0007 * CODE BY
0008 * Bruce Harrison
0009 *
0010    DEF START       DEFINE ENTRY POINT
0011    REF VWTR,KSCAN,VMBW,VMBR,VSBW,VSBR
0012 MOTBL EQU >1B00    SPRITE MOTION TABLE
0013 ATTLST EQU >1B80   SPRITE ATTRIBUTE TABLE
0014 START LWPI WS      LOAD OUR WORKSPACE
0015    LI R0,>380      POINT AT COLOR TABLE
0016    LI R1,SAVCLR AND AT STORAGE SPACE
0017    LI R2,32        32 BYTES TO GET
0018    BLWP @VMBR      READ COLOR TABLE INTO STORAGE
0019    MOV @>8370,R0 GET VDP ADDR FROM >8370
0020    LI R1,ANYKEY+1 POINT AT STORAGE BUFFER
0021    LI R2,6 SIX BYTES TO READ
0022    BLWP @VMBR      READ THOSE INTO BUFFER
0023    LI R0,>800      POINT AT CHARACTER TABLE
0024    LI R1,CHRTBL AND AT BUFFER STORAGE
0025    LI R2,256*8     256 CHARACTER DEFINITIONS
0026    BLWP @VMBR      STASH CHARACTER DEFS
0027    LI R0,>3800     SPRITE PATTERN TABLE
0028    BLWP @VMBW      WRITE CHAR DEFS THERE
0029    LI R1,BLOCK     SOLID BLOCK PATTTERN
0030    LI R2,8 EIGHT BYTES
0031    BLWP @VMBW      SPRITE CHAR 0
0032    LI R0,255*8+>3800 LAST SPRITE CHAR
0033    BLWP @VMBW      A SOLID BLOCK
0034    BL @SETBM       GO TO BIT-MAP
0035 MAGNI LI R0,>01E1 UNBLANK AND ENLARGE SPRITES
0036    BLWP @VWTR      WRITE VDP REG 1
0037    MOVB @MAGNI+3,@>38D4 PUT >E1 AT >83D4
0038 SPIRAL LI R12,20   STARTING DOT-COL 20
0039    LI R13,10       STARTING DOT-ROW 10
0040    LI R14,180      STOP ROW 180
0041    LI R15,240      STOP COLUMN 240
0042    MOV R13,R8      PUT START ROW IN R8
0043 LINE MOV R12,R7    STARTING COL IN R7
0044    AI R12,10       ADD 10 TO START COL
0045    CLR R9  COLORS BLACK ON WHITE
0046 LOOP1 BL @PLOT     DRAW ONE PIXEL
0047    INC R8  MOVE DOWN ONE ROW
0048    C R8,R14        COMPARE TO LIMIT
0049    JL LOOP1        IF LOW, REPEAT
0050    LI R9,>6000     COLOR DARK RED
0051 LOOP2 BL @PLOT     DRAW ONE PIXEL
0052    INC R7  INC COLUMN
0053    C R7,R15        COMPARE TO LIMIT
0054    JL LOOP2        IF LOW, REPEAT
0055    CLR R9  COLOR BLACK ON WHITE
0056 LOOP3 BL @PLOT     DRAW ONE PIXEL
0057    DEC R8  DEC ROW
0058    C R8,R13        COMPARE TO TOP LIMIT
0059    JH LOOP3        IF HIGH, REPEAT
0060    MOV R13,R8      PUT LIMIT IN R8
0061    AI R13,10       ADD 10 TO TOP LIMIT
0062    LI R9,>4000     COLOR DARK BLUE
0063 LOOP4 BL @PLOT     DRAW ONE PIXEL
0064    DEC R7  DEC COL
0065    C R7,R12        COMPARE TO LEFT LIMIT
0066    JH LOOP4        IF HIGH, REPEAT
0067    AI R14,-10      SUBTRACT 10 FROM LEFT LIMIT
0068    AI R15,-10      AND FROM STOP COLUMN
0069    C R13,R14       COMPARE TOP AND BOTTOM LIMITS
0070    JLT LINE        IF LESS, BACK TO START
0071    LI R8,24        ROW 24
0072    LI R7,4 COL 4
0073    LI R9,>FC00     COLOR WHITE ON DARK GREEN
0074    LI R12,SPISTR MESSAGE STRING
0075    BL @BITSTR      DISPLAY THAT
0076    LI R1,>0400     FOUR SPRITES MOVING
0077    MOVB R1,@SPMOT AT SPMOTION
0078    BL @SPRITE      SET SPRITE
0079    DATA 0  NUMBER ZERO
0080    BYTE 88,120,'A',6,-10,20 PARAMETERS (A RED "A")
0081    BL @SPRITE      SET SPRITE
0082    DATA 1  NUMBER ONE
0083    BYTE 88,120,'B',10,-10,-40 PARAMETERS (A YELLOW
"B")
0084    BL @SPRITE      SET SPRITE
0085    DATA 2  NUMBER TWO
0086    BYTE 88,120,0,3,20,20 PARAMETERS (GREEN BLOCK)
```

(See Page 16)

Successfully writing assembler programs for the Geneve 9640 — Part II

Should auld acquaintance be forgot...?

By MICHAEL ZAPF

After we learned how to write a short assembler program for the MDOS mode of the Geneve, we continue the lesson by looking at some utility routines and how to write new ones.

Let's recall some facts from the last lesson

The TASM program (the so-called 'assembler') is to convert the human-readable mnemonic code (source code) to machine-readable code (object code). The result has the well-known DIS/FIX 80 format that is used for relocatable code that can be loaded at different locations in memory. Another program called LDR is now necessary to begin execution. This one does the job of the linking loader of the Editor/Assembler when one chooses option 3 (Load and Run). But, again, the relocatability has its price: longer files and slower loading.

We find many stand-alone programs for the TI or the Geneve, i.e. they do not depend on other programs being loaded and do not offer any functionality to other programs. This, however, makes the idea of relocatable code quite useless so that most programmers tend to use absolute, fast-loading code that must be loaded by option 5 of the E/A, the memory image code.

The analogon on the Geneve is even more attractive: All programs that are to be loaded from the command line must be memory image code. The LDR program also incorporates the Save utility of the E/A: If you append another file name after all relocatable modules, the memory image of the program is saved to that file.

We learned that the first word of the first loaded module must be executable. That means that if you have only one object code module you do not need any DEF statement because no other program needs to reference the entry point. Likewise, you do not need any REF, because you cannot reference any other utility.

If that were it, it would be a torture writing a program in MDOS.

How do I write texts to the screen? That would not be so hard, but disk access? Keyboard scans? Mathemat-ical routines? The authors of MDOS obviously noticed that problem and offered all these functions inside the MDOS kernel.

How do I write texts to the screen? That would not be so hard, but disk access? Keyboard scans? Mathemat-ical routines? The authors of MDOS obviously noticed that problem and offered all these functions inside the MDOS kernel. But since we cannot reference any symbols from outside, there was no use for the acquainted BLWP @VSBW anymore.

FINALLY SOME USE FOR A COMMAND

Last month I showed you a short program that printed a string on the screen, issued a beep, and returned to the command prompt. Yes, it was done by means of XOP calls. These eXtended OPerations were already included in the TI99/4A, but disregarded.

When the CPU encounteres an XOP, it does a context switch as with the BLWP, but the vectors always stand at the same addresses: XOP 0 is at >0040, XOP 1 is at >0044, XOP 2 is at >0048 and so on. This is determined by the architecture of the TMS9900 and

(See Page 17)

ART OF ASSEMBLY —

(Continued from Page 15)

```
0087    BL @SPRITE      SET SPRITE
0088    DATA 3  NUMBER THREE
0089    BYTE 88,120,255,13,20,-40 PARAMETERS (MAGENTA
BLOCK)
0090    MOV @INTLOC,@>83C4 ENABLE USER INTERRUPT
0091    BL @KEY WAIT FOR KEYSTROKE
0092    CLR @>83C4      DISABLE USER INTERRUPT
0093    BL @SETGM       SET GRAPHICS MODE
0094 EXIT MOV @>8370,R0 GET BACK >8370 ADDRESS 0095
LI R1,ANYKEY+1 POINT AT BUFFER STORAGE
0096    LI R2,6 SIX BYTES
0097    BLWP @VMBW      WRITE THOSE BACK TO VDP
0098    LWPI >83E0      LOAD GPL WORKSPACE
0099    B @>6A  RETURN TO GPL INTERPRETER
0100 *
0101 * SUBROUTINES FOR HANDLING BIT-MAP
0102 * OPERATIONS AND TRANSITIONS
0103 *
0104 * FOLLOWING SECTION SETS COMPUTER INTO BIT-MAP MODE
0105 *
0106 SETBM LI R0,>1A0  SET FOR BLANK SCREEN
0107    BLWP @VWTR      WRITE TO VDP REG 1
0108    LI R0,>206      SET TO WRITE VDP REGISTER 2
0109    BLWP @VWTR      SIT TO >1800 (SCREEN IMAGE TABLE)
0110    LI R0,>403      SET TO WRITE TO VDP REG. 4
```

GENEVE ASSEMBLY —

(Continued from Page 14)

TMS9995. Even if the manual states that there are only XOP 1 and 2, there are in fact 16 XOPs, but only the values at (ROM) positions >0044 to >0047 are meaningful.

MDOS introduces XOP 0 as the startup for all utilities. By checking the XOP argument the operating system calls the appropriate utility routines, the largest one being number 6 with more than 50 subroutines:

```
LI    R1,VALUE
XOP   R1,0
```

Value	Service
5	Keyboard, joystick
6	Video, sound, mouse
7	Memory management
8	File access
9	Miscelleneaous utilities
10	Mathematical utilities

The respective subroutine is selected by setting R0 of the caller's workspace (the 'opcode').

SOME EXAMPLES

1. Let's do a CALL CLEAR in MDOS assembler:

We look up the opcode of the HCHAR utility; it's >2E. This routine works like the HCHAR subprogram of TI BASIC that is invoked by a CALL. So we write (assuming we have the normal 32x24 graphics mode)

```
VIDEO DATA 6
CLEAR LI   R0,>002E  * opcode
      CLR  R1        * row #
      CLR  R2        * column #
      LI   R3,32     * ASCII code of space char-
acter
      LI   R4,768    * 32 x 24 characters
      XOP  @VIDEO,0  * Video routines
```

If you always use the utility routines you will find that they really do an 'intelligent' job. For instance, they are aware of the current video mode and select the proper print routine. This is not possible if you change the video registers directly.

2. Query the mouse:

Well, that's number >31, so we use

```
VIDEO DATA 6
MOUSE LI   R0,>0031
      XOP  @VIDEO,0
```

After the call, R1 contains the current x-position, R2 the y-position. And the leftmost three bits of R3 are the buttons.

If you like, you can, in fact, define new commands. The DXOP assembler directive is supported by the TASM assembler, so you might want to write

```
DXOP SYSC,0
SYSC @VIDEO
```

But there are some caveats. If you do not pay attention, your program will most likely crash. Since you make a system call with the XOP, the system takes over control, and it has its own notions on memory allocation. As your program resides in low memory locations, it will be mapped out of visible memory. Although it is still in memory, the utility cannot access the values you set before.

In order to avoid this, put your workspace at a location where it will not be mapped out. Good chances are in >E000->FFFF, and certainly best in the on-chip RAM from >F000->F0FB. But as the XOP routines have the same problem, stay below >F080 to avoid conflicts with their WS.

Another problem sounds familiar: The more comfortable the utilities, the slower they become. If your program runs in some special context, e.g. in a windowing system, you should stay with the utilities so that the operating systems 'knows' what you are doing. If you run your self-tailored routines your program will gain speed, but most likely not run correctly inside other programs.

AND WE'LL DO IT AFTER ALL

The TASM and LDR normally come with some documentation that list the available XOPs. Therefore, I don't want to spend too much space and time on this because we have some more important issues to discuss.

The Achilles' heel is — again — the video access. Imagine you have some very time-critical portions in your code that do some dumb job, e.g. write some single bytes to video RAM. This was the case with my 'Fractals!' program. I simply could not spend so much time in calling the XOP routines for the video access because there are many such simple accesses. Anyway, those routines do not reach every location in the VDP RAM. So it is important to know how to write a routine that is

1. fast
2. allows access to ALL memory locations.

VDP access is the same for TI's TMS9918 as it is for the V9938 inside the Geneve. (Most of the available video extension cards for the TI also use the V9938.) There is no direct memory access (DMA) but a set of ports that the CPU writes to or reads from. In addition, there are some control registers and status registers.

MAP OF VDP PORTS

Purpose	TI-99/4A	Geneve
Read byte	>8800	>F100
Write byte	>8C00	>F100
Address/Register	>8C02	>F102
Status	>8802	>F102
Palette	—	>F104
Indirect Register	—	>F106

Note that read and write accesses use the same port on the Geneve. The RAM address is set by first writing the low order byte, then the least significant five bits of the high order byte, OR-ed with >40 to initiate a write. If we need to specify further bits of the address (maximum 17 bit for 128 KB RAM), we have to load the VDP register 14 with these bits. Attention: The VR14 keeps its value. You need to reset it for further accesses in other memory areas. On the other hand, if you run over a >4000 multiple, the contents of VR14 are automatically increased.

In order to set VR14, you have to write some kind of VWTR routine. It does the following: Write the value to >F102, then write the register number, OR-ed with >80, to >F102.

I will conclude this lesson with a code fragment that you may fit into your own programs. It is a generic video access routine that offers the functionality of the known VDP access routines of

(See Page 18)

GENEVE ASSEMBLY —

(Continued from Page 17)

the TI, extended to the large 128 KB memory. If you put the routines in a separate module, do not forget to DEFine them so that your main program can REFerence them. I hope that the comments will be helpful in understanding the way the routine works.

You can get in contact with me in three electronic ways: Write an E-Mail to "zapf@vsb.informatik.uni-frankfurt.de". 2. Write an E-Mail to the TI List Server if you have already subscribed. 3. Post a message in the USENET to "comp.sys.ti".

```
*
* Video routines -- Michael Zapf, Oct. 1996 *
* All but the last must be called like this: *
*      BLWP @routine
*      DATA x
*
* with x = >0000 or x = >0400 (17th bit of the address) *
* Caller's workspace:
*
* R0 = VDP RAM address (full 16 bit!)
* R1 = CPU RAM address
* R2 = Byte count (only for multiple byte access) *
* The last routine is call [illegible]th
*
*      LI   R0,xy
*      BLWP @VWTRU
*
*      with x = register number
*           y = register value
*

VIDWS   BSS  32              private video workspace
VSBWU   DATA VIDWS,VSBW      single byte write
VSBRU   DATA VIDWS,VSBR      single byte read
VMBWU   DATA VIDWS,VMBW      multiple bytes write
VMBRU   DATA VIDWS,VMBR      multiple bytes read
VWTRU   DATA VIDWS,VWTR      write to register

VSBW    BL   @VIDADW         set for writing and get byte
        MOVB R1,@>F100       write byte to port
        RTWP
VSBR    BL   @VIDADR         set for reading
        MOVB @>F100,@2(R13)      read byte from port,
write to R1(caller)
        RTWP
VMBW    BL   @VIDADW
        JEQ  VMBWE           no bytes?
VMBWL   MOVB *R1+,@>F100     write to port
        DEC  R2              until finished
        JGT  VMBWL
VMBWE   RTWP
VMBR    BL   @VIDADR
        JEQ  VMBWE
VMBRL   MOVB @>F100,*R1+
        DEC  R2
        JGT  VMBRL
        RTWP
VWTR    MOV  *R13,R0         Get R0(caller)
        LIMI 0               Block interrupts
        ORI  R0,>8000        Set VDP Register write flag
bit
        SWPB R0
        MOVB R0,@>F102       Write low order byte
        SWPB R0
        MOVB R0,@>F102       Write high order byte
        RTWP

VIDADW LI    R2,>4000        Set VDP RAM write flag
       JMP   $+4
VIDADR CLR   R2              Clear it
       LIMI  0
       MOV   *R13,R0         Get R0(caller) = VDP RAM ad-
dress
       MOV   R0,R1
       ANDI  R1,>3FFF        Mask it (address range =
16KB)
       SRL   R0,6            Get the leftmost two bits
       ANDI  R0,>0300
       A     *R14+,R0        Add offset (DATA 0 or >0400)
       MOVB  R0,@>F102       Set the value
       LI    R0,>8E00        in VDP Register 14
       MOVB  R0,@>F102
       SOC   R2,R1           Put the flag bit into the ad-
dress
       SWPB  R1
       MOVB  R1,@>F102
       SWPB  R1
       MOVB  R1,@>F102       Address is set
       MOV   @2(R13),R1      Fetch the values from the
       MOV   @4(R13),R2      caller's workspace
       RT
```

Precise math

When it comes to big numbers, computers have their limits

By JIM PETERSON

The TI99/4A has long been renowned for its mathematical accuracy, far greater than some computers costing many times as much. I understand that this is because of its 16-bit, rather than 8-bit processor. In fact, the TI99/4A is far more accurate than is necessary for any home computer use or business use, or for most any scientific use, and this unnecessary degree of accuracy is one of the reasons for the slowness of this computer.

Also, when dealing with very large numbers or numerous decimal places, the actual computations are far more accurate than the screen display might indicate. See section II-66 of the User's Reference Guide.

However, even the TI99/4A has its limitations. Numbers are stored internally in a system called radix-100 (see the User's Reference Guide section II-9); for a full explanation of this system, see the Editor/Assembler manual. Or, for a quick look, run the program RADIX/DEMO. This saves a number to disk in internal format, reads it back as an ASCII string, and examines it byte by byte. Try inputting 123456789. You will find that the first ASCII controls the length of the number, its decimal point, whether it is positive or negative, etc., and the remaining 7 ASCII show the decimal digits.

Since this format always consists of 8 bytes, of which only 7 represent the digits, and each byte represents at most two digits, the TI99/4A can only accurately maintain a number less than 15 digits long.

Try inputting 100 trillion, which is 100,000,000,000,000 or in computerese 100000000000000, 1 followed by 14 zeros, therefore more than 14 digits long. Now try 100000000000001 — the radix —100 representation is exactly the same! Try 100000000000049 — still the same!

Does this mean that the computer is no longer accurate? In command mode, enter PRINT 100000000000000-99999999999999 That is 1 followed by 14 zeros minus 14 nines. The answer should be 1 and it is. Hit function 8 and change that first figure to 100000000000049. The answer now should be 51 — but it is still

It was necessary to teach the computer to compute in the old-fashioned pencil and paper way — multiply digit by digit from right to left, recording the rightmost digit of each result and carrying over the leftmost, if any, to be added to the next computation.

1! When a number is more than 14 digits long, it is represented in "scientific notation," which may be scientific but is certainly not accurate, and is maintained in the computer as a base number multiplied by 10 to a power of 10 — which means that it is rounded off to a string of zeros! I became interested in this when the TI*MES newsletter of the English users group published a challenge — write a program in any language to find the lowest power of 7 which contains six 7s in sequence. I know very little about math, but I like challenges, so I wrote a "tinygram" one-screen solution in XBASIC, which was published in Tips From The Tigercub No. 60.

The only way to solve this, presumably, is to repeatedly multiply by 7 and check the result for a string of six 7s. However, since the answer is more than 14 digits long, it would be in scientific notation, which could not be searched by POS — in fact, as it turns out, the answer is greater than the TI99/4A can handle even in radix-100!

So, it was necessary to teach the computer to compute in the old-fashioned pencil and paper way — multiply digit by digit from right to left, recording the rightmost digit of each result and carrying over the leftmost, if any, to be added to the next computation. To get around the 14-digit limitation, the result is recorded as a string, which can be up to 255 digits long, and converted to numeric one digit at a time. The result is the filename SEVENS.

Having gotten interested, I went ahead and wrote similar routines to multiply any two numbers of any length (providing the result is not over 255 digits), including decimals and negatives, with absolute accuracy, and another to add up any number of values of any length, within the same limitation. These were also published in Tips No. 60. The addition routine would only handle integers, but the improved version below can also handle decimals. I have now also written a routine for subtraction. These are programs named MULT, ADD and SUB. Division would be considerably more difficult, and I decided not to spend time on it, since all this is really just an interesting programming exercise with little practical value.

RADIX/DEMO

```
100 OPEN #1:"DSK1.TEST",INTE
RNAL,OUTPUT
110 INPUT N
120 PRINT N
130 PRINT #1:N
140 CLOSE #1
150 OPEN #1:"DSK1.TEST",INTE
RNAL,INPUT
160 INPUT #1:N$
170 PRINT LEN(N$)
180 FOR J=1 TO LEN(N$):: PRI
NT ASC(SEG$(N$,J,1));:: NEXT
J
190 PRINT
200 CLOSE #1
210 GOTO 100
```

SEVENS

```
1 DISPLAY AT(3,9)ERASE ALL:"
SIX SEVENS":"":"  by Jim
Peterson":"":" This tinygra
```

(See Page 20)

PRECISE MATH —

(Continued from Page 19)

```
m was written in response to
a challenge in the TI*MES
newsletter of"
2 DISPLAY AT(10,1):"England,
to find the lowest power of
7 which contains six 7's
in sequence. The TI cannot s
olve this by normal"
3 DISPLAY AT(14,1):"means, s
o it must be taught the old-
fashioned method of multipli
cation.":" The program takes
about 24 minutes to find th
e answer."
100 A$=STR$(7):: Y=1 ! the n
umber to be multiplied is or
iginally set at 7, in string
format, and the power is 1
110 Y=Y+1 :: FOR J=LEN(A$)TO
1 STEP -1 :: E=(VAL(SEG$(A$
,J,1))*7+X)/10 ! the counter
of powers is incremented by
1 each that execution goes
111 ! thru this line. The J
loop goes thru the A$ string
from right to left, as mult
iplication is done on paper,
converting a character to
112 ! numeric format, multip
lying it by 7, adding any ca
rryover quantity, then divid
ing by 10 to make the rightm
ost digit a decimal
120 X=INT(E):: F=(E-X)*10 ::
X$=STR$(F)&X$ :: NEXT J ! t
he INT function strips off t
he decimal part, leaving the
carryover quantity; this
121 ! integer is subtracted
from the original value to o
btain the decimal portion, w
hich is multiplied by 10 to
become an integer, converted
122 ! to a string character
and added to the left of the
string X$
130 IF X>0 THEN X$=STR$(X)&X
$ ! when the loop is complet
ed, if there is a carryover
quantity it is added to the
left of the result, just as
131 ! in paper and pencil fi
guring
140 IF POS(X$,"777777",1)<>0
THEN 160 ! POS is used to f
ind if the string contains t
he six 7's yet; if so, jump
over to print it
150 A$=X$ :: X$="" :: X=0 ::
GOTO 110 ! otherwise, this
result becomes the new value
to be multiplied by 7, the
add-on string X$ is made a
151 ! null string, the carry
over value is zeroed, and ex
ecution returns to start ove
r
160 PRINT "7^";STR$(Y);"=";X
$
```

ADD

```
100 DISPLAY AT(3,7)ERASE ALL
:"PRECISE ADDITION":"":"
by Jim Peterson" ! Revis
ed 91/1/21
110 DISPLAY AT(6,7):" This
program will total upto 100
numbers (change the":"DIM fo
r even more) of any"
120 DISPLAY AT(10,1):"length
(up to 255 digits),":"inclu
ding decimals, to any":"tota
l of not over 255 char-":"ac
ters, without rounding off"
130 DISPLAY AT(14,1):"decima
ls or using scientificnotati
on."
140 DIM C$(100)! change this
to add up even more numbers
150 DISPLAY AT(20,1):"Input
from K":" (D)isk or":" (K)
eyboard?" :: ACCEPT AT(20,12
)VALIDATE("DK")SIZE(-1):Q$ :
: IF Q$="K" THEN 190
160 DISPLAY AT(12,1)ERASE AL
L:"Filename? DSK" :: ACCEPT
AT(12,14):F$ :: OPEN #1:"DSK
"&F$,INPUT
170 X=X+1 :: INPUT #1:C$(X):
: GOSUB 360 ! X counts size
of array; this reads a numer
ic file; if LINPUT was used,
each string would have
171 ! ASCII 32 before and af
ter it, which would cause pr
oblems!
180 IF EOF(1)<>1 THEN 170 EL
SE CLOSE #1 :: GOTO 220
190 DISPLAY AT(12,1)ERASE AL
L:"Press ENTER when finished
":"":""
200 X=X+1 :: INPUT C$(X)! ag
ain, X counts size of array
210 IF C$(X)<>"" THEN GOSUB
360 :: GOTO 200 ELSE X=X-1 !
if Enter is pressed, the st
ring is a null string which
is not used so reduce the
211 ! count by 1
220 FOR J=1 TO X :: IF PP=0
THEN 270 ! see subroutine in
360; if no value contained
a decimal, PP=0
230 Y=POS(C$(J),".",1)! othe
rwise, find position of deci
mal in each element of array
240 IF Y=0 THEN C$(J)=C$(J)&
RPT$("0",PP):: GOTO 270 ! if
it has no decimal, add the
number of 0's equal to the v
alue of PP
250 Y=LEN(C$(J))-Y :: IF Y=P
P+1 THEN 260 :: C$(J)=C$(J)&
RPT$("0",PP-Y)! otherwise, f
ind the distance of the deci
mal from the right end;
251 ! if it is the 1st chara
cter, no trailing 0's are ne
cessary; otherwise, add enou
gh 0's so that all decimals
are lined up
260 Y=POS(C$(J),".",1):: C$(
J)=SEG$(C$(J),1,Y-
1)&SEG$(C$
(J),Y+1,255)! find the new p
osition of the decimal; take
the string apart and
261 ! reassemble it to remov
e the decimal
270 M=MAX(M,LEN(C$(J))):: NE
XT J ! then measure the leng
th of the string, compare it
with the previous value of
M; if greater, assign that
271 ! new value to M, which
ends up having the value of
the longest string in the ar
ray
280 FOR J=1 TO X :: IF LEN(C
$(J))<M THEN C$(J)=RPT$("0",
M-LEN(C$(J)))&C$(J)! go thru
the array again; if a strin
```

(See Page 21)

PRECISE MATH —

(Continued from Page 20)

```
g is shorter than the
281 ! maximum, add enough 0'
s at the left so all strings
are of equal length
290 NEXT J :: FOR J=M TO 1 S
TEP -1 :: FOR K=1 TO X
300 G=G+VAL(SEG$(C$(K),J,1))
! the J loop goes thru the s
trings from right to left; t
he K loop goes thru the arra
y from top to bottom;
301 ! each character is conv
erted to its value; these ar
e totaled up just as additio
n is done with pencil and pa
per
310 NEXT K
320 G=(G+H)/10 :: L=INT(G)::
G=(G-L)*10 :: D$=STR$(G)&D$
:: H=L :: G=0 :: NEXT J ! a
ny carryover quantity is
321 ! added to the total whi
ch is then divided by 10 to
make the rightmost digit a d
ecimal; the decimal portion
is stripped off to leave an
322 ! integer; the integer i
s subtracted from the total
to find the decimal portion,
which is converted back to
an integer by multiplying by
323 ! 10, then converted to
a string character and added
at the left of the D$ strin
g; the integer L becomes the
new carryover value, G is
324 ! zeroed out before addi
ng up the next column
330 IF H>0 THEN D$=STR$(H)&D
$ ! when finished, if any ca
rryover quantity remains it
is converted to a string and
added at the left of the
331 ! total, just as with pa
per and pencil
340 IF PP>0 THEN D$=SEG$(D$,
1,LEN(D$)-PP)&"."&SEG$(D$,LE
N(D$)-PP+1,255)! if there wa
s any value containing a dec
imal, the decimal is
341 ! inserted at the proper
place in the answer
350 DISPLAY AT(12,1)ERASE AL
L:"The exact total is":D$:""
:"The computer would normall
y total this as":STR$(T):: S
TOP
355 DISPLAY AT(20,1):"Anothe
r? (Y/N)" :: ACCEPT AT(20,16
)SIZE(1)VALIDATE("YN"):Q$ ::
IF Q$="N" THEN STOP ELSE CA
LL CLEAR :: GOTO 150
360 T=T+VAL(C$(X)):: P=POS(C
$(X),".",1):: IF P>0 THEN P=
LEN(C$(X))-P :: PP=MAX(PP,P)
:: RETURN ELSE RETURN
361 ! each input of values,
from eith disk or keyboard,
goes to this subroutine; the
ir values are totaled in T b
y normal computer means, to
362 ! be printed out in 350;
a check is made for the posi
tion of a decimal; if other
than 0, a decimal does exist
; the distance of the
363 ! decimal from the right
end is determined; MAX is u
sed to find the longest deci
mal value in the array. Note
the ELSE RETURN; if this is
364 ! not present, program e
xecution would drop out of l
ine 360 and terminate when t
he first integer was found.
```

MULT

```
100 DIM C$(100)!array to hol
d interim calculations - can
be increased to memory limi
tations
110 DISPLAY AT(12,1)ERASE AL
L:"FIRST NUMBER?" :: ACCEPT
AT(14,1)VALIDATE(NUMERIC)BEE
P:A$ ! accept first number a
s a string, therefore can be
111 ! of any length without
going into scientific notati
on; for numbers over 28 digi
ts long, change this to INPU
T
120 IF SEG$(A$,1,1)="-" THEN
A$=SEG$(A$,2,255):: M=1 !if
the first character is the
minus sign, remove it, but s
et the flag M to 1
130 A=LEN(A$):: D1=POS(A$,".
",1):: IF D1>0 THEN A$=SEG$(
A$,1,D1-1)&SEG$(A$,D1+1,255)
:: D1=A-D1 ! measure the len
gth of the string; find the
131 ! position of the decima
l; if it is more than 0 (dec
imal is present) reassemble
the string to remove it and
change D1 by subtracting it
132 ! from the string length
, to indicate the number of
decimal places to be counted
off from the right in the a
nswer
140 DISPLAY AT(16,1):"SECOND
NUMBER?" :: ACCEPT AT(18,1)
VALIDATE(NUMERIC)BEEP:B$ ! d
o the same for the 2nd numbe
r
150 IF SEG$(B$,1,1)="-" THEN
B$=SEG$(B$,2,255):: M=M+1 !
and again the same; if negat
ive, increase value of M by
1
160 Y=LEN(B$):: D2=POS(B$,".
",1):: IF D2<>0 THEN B$=SEG$
(B$,1,D2-1)&SEG$(B$,D2+1,255
):: D2=Y-D2 :: D1=D1+D2 :: Y
=Y-1
161 ! again the same; add D2
to D1 to record total numbe
r of spaces that decimal is
to be offset; adjust Y (leng
th of string) to reflect
162 ! deletion of decimal, b
ecause Y is used in next lin
e
170 FOR J=Y TO 1 STEP -1 ::
W=W+1 :: B=VAL(SEG$(B$,J,1))
:: FOR K=LEN(A$)TO 1 STEP -1
:: A=VAL(SEG$(A$,K,1))
171 ! the J loop goes thru t
he 2nd string from right to
left character by character,
converting each to its nume
ric value; W is the counter
172 ! for these. The K loop
goes thru the 1st string fro
m right to left in the same
way
173 ! multiplication is perf
ormed, digit by digit, in th
e way it would be done with
paperer and pencil -
180 D=(A*B+X)/10 ! the two d
igits are multiplied and the
```

(See Page 22)

PRECISE MATH —

(Continued from Page 21)

```
carryover value X (always 0
for the rightmost digit of
A$) is added; divided by 10
181 ! to make the rightmost
digit a decimal
190 E=INT(D):: F=(D-E)*10 ::
C$(J)=STR$(F)&C$(J):: X=E :
: NEXT K ! the decimal porti
on is dropped to find the ca
rryover value for X;
191 ! the decimal portion is
extracted and multiplied by
10 to convert it to an inte
ger which is converted to a
string which is added to the
192 ! left of the C$ array e
lement which is completed wh
en the K loop has scanned th
e entire first number
200 IF X>0 THEN C$(J)=STR$(X
)&C$(J)! if a carryover quan
tity remains, it is converte
d to a string and added to t
he left of the C$ element
210 C$(J)=C$(J)&RPT$("0",W-1
)! the C$ array corresponds
to the values which are list
ed in a pencil and paper cal
culation, each of which is
211 ! offset one space to th
e left; this is accomplished
by using the W counter to a
dd one 0 to the end of the s
tring for each calculation
212 ! afer the first
220 X=0 :: NEXT J ! the carr
yover value X is reduced to
0 before going to the next d
igit of the second number
230 L=LEN(C$(1)):: FOR J=1 T
O Y :: L2=LEN(C$(J)):: IF L2
<L THEN C$(J)=RPT$("0",L-L2)
&C$(J)
231 ! the length of C$(1), w
hich is the last one calcula
ted, is measured; the J loop
goes thru the array, measur
ing each element; if it is
232 ! less than the first, e
nough zeros are added to its
left to make all strings in
the array the same length;
this simplifies the adding
233 ! up in the next step
240 NEXT J
250 FOR J=LEN(C$(1))TO 1 STE
P -1 :: FOR K=1 TO Y :: G=G+
VAL(SEG$(C$(K),J,1)):: NEXT
K ! the J loop goes thru the
array; the K goes thru each
251 ! element of the array f
rom left to right, convertin
g each character to its valu
e and accumulating their tot
al in G
260 G=(G+H)/10 :: L=INT(G)::
G=(G-L)*10 :: D$=STR$(G)&D$
:: H=L :: G=0 :: NEXT J ! t
he carryover value (always 0
to start with) is added to
261 ! the total, which is di
vided by 10 to make the righ
tmost digit a decimal; the i
nteger is taken to drop the
decimal portion and find the
262 ! carryover quantity H;
the decimal quantity is extr
acted and converted to an in
teger by multiplying by 10
263 ! and converted to a str
ing which is added to the le
ft of the string D$ :: the v
alue of G is zeroed out befo
re going to the next element
270 IF H>0 THEN D$=STR$(H)&D
$ ! when the J loop is compl
eted, any remaining carryove
r quantity is converted to a
string and added at the
271 ! left of the string D$
280 IF D1>0 THEN D$=SEG$(D$,
1,LEN(D$)-D1)&"."&SEG$(D$,LE
N(D$)-D1+1,255)! if D1 is mo
re than 0 (one or both numbe
rs contained decimals - see
281 ! lines 130 and 160) the
string D$ is separated and
a decimal is inserted at the
point where the value of D1
indicates it should be
290 IF M=1 THEN D$="-"&D$ !
if M=0 then both numbers wer
e positive, or if M=2 then b
oth were negative and their
multiplication would give a
291 ! positive result, but i
f M=1 (see 120 and 150) then
a minus sign is added at le
ft
300 PRINT D$ ! and the resul
ts of the computation are di
splayed.
```

SEVENS

```
1 DISPLAY AT(3,9)ERASE ALL:"
SIX SEVENS":"":"  by Jim
Peterson":"":" This tinygra
m was written in response to
a challenge in the TI*MES
newsletter of"
2 DISPLAY AT(10,1):"England,
to find the lowest power of
7 which contains six 7's
in sequence. The TI cannot s
olve this by normal"
3 DISPLAY AT(14,1):"means, s
o it must be taught the old-
fashioned method of multipli
cation.":" The program takes
about 24 minutes to find th
e answer."
100 A$=STR$(7):: Y=1
110 Y=Y+1 :: FOR J=LEN(A$)TO
1 STEP -1 :: E=(VAL(SEG$(A$
,J,1))*7+X)/10
120 X=INT(E):: F=(E-X)*10 ::
X$=STR$(F)&X$ :: NEXT J
130 IF X>0 THEN X$=STR$(X)&X
$
140 IF POS(X$,"777777",1)<>0
THEN 160
150 A$=X$ :: X$="" :: X=0 ::
GOTO 110
160 PRINT "7^";STR$(Y);"=";X
$
```

SUB

```
100 DISPLAY AT(3,5)ERASE ALL
:"PRECISE SUBTRACTION":" ":"
     by Jim Peterson"
110 DISPLAY AT(7,1):"  Will
perform subtraction  on any
two numbers of any   length
(for longer numbers  change
the ACCEPTs to"
120 DISPLAY AT(11,1):"INPUTs
) of up to 255 digits and gi
ve an exact result of up to
255 digits without    roundi
ng off decimals or"
130 DISPLAY AT(15,1):"using
scientific notation."
```

(See Page 23)

PRECISE MATH —

(Continued from Page 22)

```
140 D$="" :: DISPLAY AT(20,1
):"First number?" :: ACCEPT
AT(21,1)VALIDATE(NUMERIC):A
$
 ! the add-on string D$ is r
eset to a null string
141 ! because program execut
ion returns here from 290. F
or even longer numbers chang
e ACCEPT ATs to INPUTs
150 DISPLAY AT(22,1):"Second
 number?" :: ACCEPT AT(23,1)
VALIDATE(NUMERIC):B$ :: T=VA
L(A$)-VAL(B$):: FLAG=0 ! dit
to. T is the result of the
151 ! computer's attempt to
solve the problem directly
160 IF (VAL(B$)>VAL(A$))OR(V
AL(A$)=VAL(B$)AND B$>A$)THEN
 T$=A$ :: A$=B$ :: B$=T$ ::
FLAG=1 ! if 2nd number is la
rger than 1st, swap them and
161 ! set flag that result w
ill be negative. This also a
ttempts to take care of the
situation where the true val
ue of B$ is just slightly
162 ! more than A$ but the C
omputer's VAL of the two is
the same due to the inaccura
cy of scientific notation. T
his may not always work
170 AL=LEN(A$):: AP=POS(A$,"
.",1):: IF AP>0 THEN AX=AL-A
P ! find the length of A$, t
he position of the decimal i
n A$ from the left, and its
position from
171 ! the right
180 BL=LEN(B$):: BP=POS(B$,"
.",1):: IF BP>0 THEN BX=BL-B
P ! ditto for B$
190 M=MAX(AX,BX):: IF M=0 TH
EN 220 :: IF AX=BX THEN 200
:: IF AX<BX THEN A$=A$&RPT$(
"0",BX-AX)ELSE B$=B$&RPT$("0
",AX-BX)
191 ! M is the maximum numbe
r of decimal spaces found. I
f it is 0 (no decimals) skip
 over the decimal adjusting
routine. If both numbers
192 ! have same number of de
cimal places, skip over the
lining-up routine. Otherwise
 add zeros to the right of t
he shorter number to line up
193 ! the decimals
200 IF AP<>0 THEN A$=SEG$(A$
,1,AP-1)&SEG$(A$,AP+1,255)!
if A$ had a decimal, reassem
ble the string to remove it
210 IF BP<>0 THEN B$=SEG$(B$
,1,BP-1)&SEG$(B$,BP+1,255)!
ditto if B$ has a decimal
220 AL=LEN(A$):: BL=LEN(B$):
: IF AL>BL THEN B$=RPT$("0",
AL-BL)&B$ ! find the new len
gths of both strings; if A$
is longer, add 0's in front
221 ! of B$ to make them equ
al
230 FOR J=AL TO 1 STEP -1 ::
 A=VAL(SEG$(A$,J,1))-F :: B=
VAL(SEG$(B$,J,1)):: IF A<B T
HEN A=A+10 :: F=1 ELSE F=0
231 ! the J loop goes thru b
oth strings from right to le
ft, converting single charac
ters to values; if A has bee
n "borrowed from" F's value
232 ! of 1 is subtracted. If
 A is then less than B, 10 i
s added to A and F is set to
 1 to indicate borrowing fro
m the digit to right, else
233 ! F is reset to 0. This
is the same as pencil and pa
per subtraction
240 C=A-B :: D$=STR$(C)&D$ :
: NEXT J ! the lower number
is subtracted from the upper
, the result converted to a
string character and added
241 ! to the left of the str
ing D$. When all is done -
250 IF M>0 THEN M=LEN(D$)-M
:: D$=SEG$(D$,1,M)&"."&SEG$(
D$,M+1,255)! if the maximum
number of decimal places was
 more than 0,
251 ! (i.e., there was a dec
imal), the D$ string is reas
sembled to put the decimal i
n its proper place
260 IF SEG$(D$,1,1)="0" THEN
 D$=SEG$(D$,2,255):: GOTO 26
0 ! to strip the superfluous
 zeros from the right; if th
e first digit is a zero,
261 ! make the string consis
t of its 2nd to last charact
ers (using 255 is the same,
and easier, than specifying
the actual length). The line
262 ! goes back to itself un
til a non-zero is found
270 IF FLAG=1 THEN D$="-"&D$
 ! if the flag was set to !
in 160, the minus sign is ad
ded to the front of D$
280 DISPLAY AT(12,1)ERASE AL
L:"The exact result is":D$:"
 ":"The normal computer resu
lt  is":STR$(T)
290 DISPLAY AT(20,1):"Anothe
r? (Y/N)" :: ACCEPT AT(20,16
)SIZE(1)VALIDATE("YN"):Q$ ::
 IF Q$="N" THEN STOP ELSE CA
LL CLEAR :: GOTO 140
```

HSPGL card is upgraded

Gerd Weissmann advises that his HSGPL Card has been upgraded to HSGPL II. He says it loads all Milton Bradley modules inclusively and has new hardware and software improvements.

Weissmann also says the new CPU Card is now available with 16-bit bus. It requires the HSGPL Card and any 80-column card. His EVPC Card is recommended. The CPU Card is available in the following configurations (all are 16-bit, prices are in U.S. funds):

32KB RAM	8 KB ROM	$200
32KB RAM	32 KB ROM	$215
256 KB RAM	64 KB ROM	$355
1 MB RAM	64 KB ROM	$430

Add US $45 for overseas delivery (more when received and tested). For information or to order, contact Gerd Weissmann, Koenigstr. 17-19, D-67655 Kaiserslautern, Germany, tel./fax 0631/12169.

GRIDLUCK

Pick your grid and take your chances

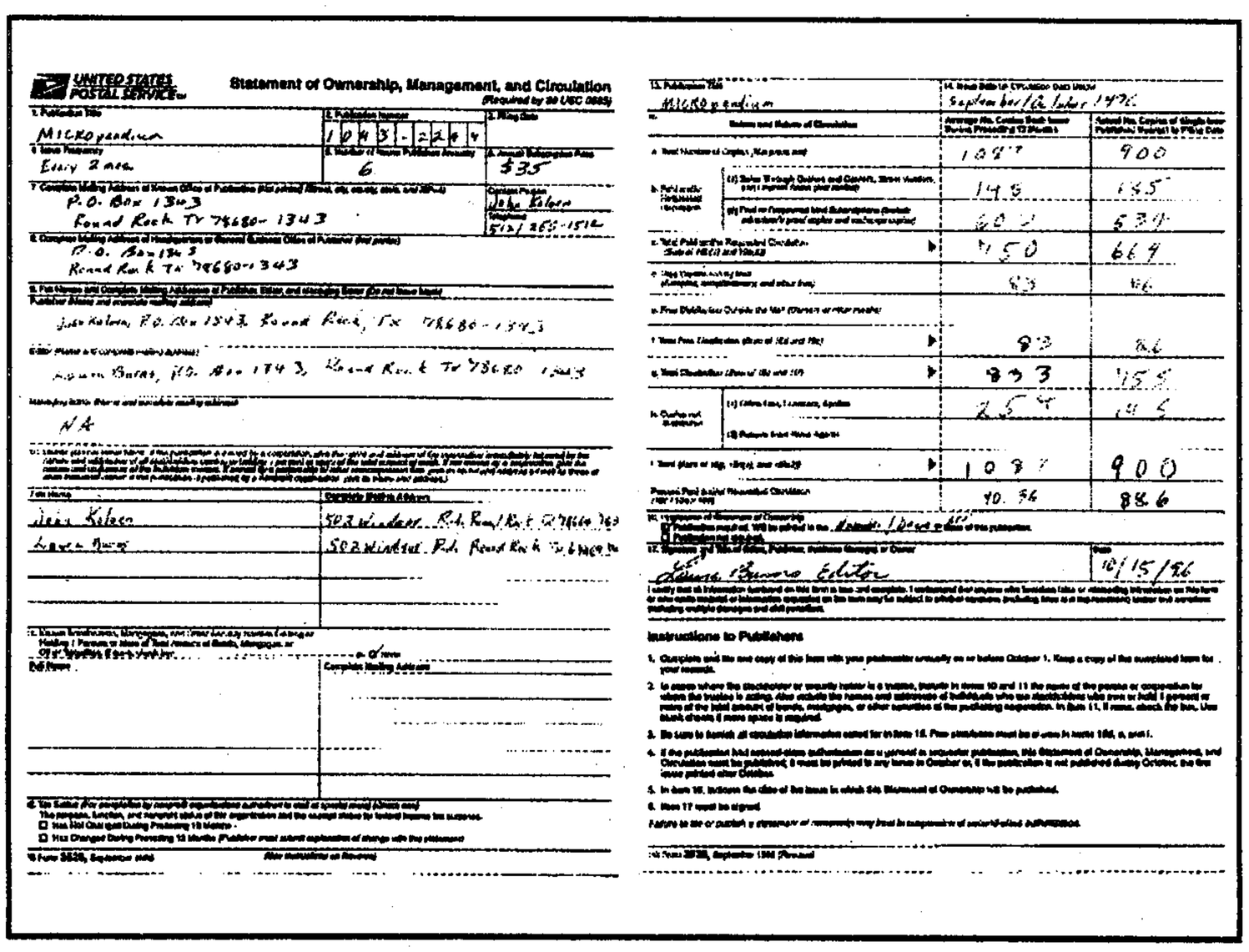
BY DON STEFFEN

Gridluck is a game of chance. This Extended BASIC program first asks for the size grid you want. Your choices are 4x4, 5x5, 6x6, 7x7, or 8x8. Then it asks you to pick four numbers, ranging from one to the maximum number for the grid. After you enter your numbers, the program fills the grid randomly with one of each number for the grid and puts a dollar sign in each square that has one of your numbers in it. It then calculates a winner. You win if your numbers are placed in the four corners of the grid, or if your four numbers are in a square next to each other, or if your numbers are lined up horizontally, vertically, or diagonally, like Tic-Tac-Toe.

GRIDLUCK

```
1 ! CHANGE LINES 16 & 55 TO
THOSE ABOVE TO GIVE YOURSELF
 A BETTER CHANCE!
2 ! THEN TRY TO WATCH THE SP
RITE TO PICK YOUR NUMBERS
8 ! ** GRIDLUCK PROGRAM ***
IT NEVER SELECTS SAME NUMBER
 TWICE. BY DON STEFFEN INSPI
RED BY MICROPENDIUM  DEC95 R
ANDOMIZER
10 DIM D$(64),WP(8,8),P(4)::
 CALL SCREEN(3):: DISPLAY ER
ASE ALL:"   RANDOMIZER " ::
FOR X=5 TO 8 :: CALL COLOR(X
,16,1):: NEXT X
11 CALL CHAR(91,"18181818181
81818",92,"181818FFFF181818
"
,95,"000000FFFF",64,"101838
F
E381810")):: GOTO 55
12   A$=SEG$("#$%&'()*+,-
./012
3456789:;<=>?@ABCDE-
FGHIJKLMN
OPQRSTUVWXYZ[\]^_`abcde-
fghij
klmnopq",1,N^2):: DISPLAY AT
(24,7):"FILLING ARRAY"
13 CALL SCREEN(2):: FOR J=1
TO N^2 :: RANDOMIZE :: Y=INT
(RND*LEN(A$)+1)::
X=ASC(SEG$
(A$,Y,1))-34 :: A$=SEG$(A$,1
,Y-1)&SEG$(A$,Y+1,LEN(A$))
14 ! E=E+(E>12)*E+RND*4+1 ::
 DISPLAY AT(24,E):"FILLING A
RRAY" ! OPTIONAL DISPLAY15 C
A                L           L
HCHAR(RND*20+1,RND*28+3,
64):: IF J=N OR J=N^2-N+1 TH
EN CALL SOUND(10,2500,5)
16 X$=STR$(X):: D$(J)=X$ ::
CALL PATTERN(#1,X+64):: NEXT
 J :: CALL SCREEN(3)
17 IF N=52 THEN 24 ELSE IF N
=4 OR N=5 OR N=6 OR N=7 OR N
=8 THEN 30 ELSE 27
27 PRINT "RANDOMIZED" :: FOR
 X=H*C TO N^2 :: PRINT D$(X)
;"  ";:: NEXT X
28 INPUT "DO AGAIN? Y/N ":Z$
 :: IF Z$="Y" THEN CALL CLEA
R :: GOTO 12 ELSE STOP
29 ! ~~~~~ PICK 4 GRIDLUCK
  MAKE GRID
30 Q$="PICK 4 NUMBERS FROM 1
 TO  "&STR$(N^2):: CALL PATT
ERN(#1,32):: INPUT Q$:P(1),P
(2),P(3),P(4):: G=N :: V=7+(
G>4)*3+(G>6)*3
31 H=11+(G>5)*3+(G>7)*3
32 CALL CLEAR :: FOR X=V TO
V+G*3+(G=8)STEP 3 :: CALL HC
HAR(X,H,95,G*3):: NEXT X ::
FOR Y=H TO H+G*3 STEP 3 :: C
ALL VCHAR(V,Y,91,G*3)
33 NEXT Y :: FOR X=V TO V+G*
3+(G=8)STEP 3 :: FOR Y=H TO
H+G*3 STEP 3 :: CALL HCHAR(X
,Y,92):: NEXT Y :: NEXT X
34 ! ~~~~~ FILL GRID
35 Z=1 :: FOR X=V+2 TO V+G*3
-1 STEP 3 :: VP=VP+1 :: FOR
Y=H+1 TO H+G*3 STEP 3 :: HP=
HP+1
36 R=VAL(D$(Z)):: Z=Z+1 :: E
=INT(R/10):: CALL HCHAR(X,Y,
E+48):: CALL HCHAR(X,Y+1,R-E
*10+48)
37 FOR C=1 TO 4 :: IF P(C)=R
 THEN WP(VP,HP)=1 :: CALL HC
HAR(X-1,Y,36,2):: GOTO 39 EL
```

(See Page 25)



UNITED STATES POSTAL SERVICE — Statement of Ownership, Management, and Circulation (Required by 39 USC 3685)

1. Publication Title	2. Publication Number	3. Filing Date
MICROpendium	1 0 4 3 - 2 2 4 9	
4. Issue Frequency: Every 2 mos.	5. Number of Issues Published Annually: 6	6. Annual Subscription Price: $35

7. Complete Mailing Address of Known Office of Publication (Not printer): P.O. Box 1343, Round Rock TX 78680-1343 — Contact Person: John Kilcoyne; Telephone: 512/255-1512

8. Complete Mailing Address of Headquarters or General Business Office of Publisher (Not printer): P.O. Box 1343, Round Rock TX 78680-1343

9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor (Do not leave blank)
Publisher (Name and complete mailing address): John Kilcoyne, P.O. Box 1343, Round Rock, TX 78680-1343
Editor (Name and complete mailing address): Lauren Burns, P.O. Box 1343, Round Rock TX 78680-1343
Managing Editor (Name and complete mailing address): N/A

10. Owner (If the publication is owned by a corporation, give the name and address of the corporation immediately followed by the names and addresses of all stockholders owning or holding 1 percent or more of the total amount of stock. If not owned by a corporation, give the names and addresses of the individual owners. If owned by a partnership or other unincorporated firm, give its name and address as well as those of each individual owner. If the publication is published by a nonprofit organization, give its name and address.)

Full Name	Complete Mailing Address
John Kilcoyne	502 Windsor Rd, Round Rock TX 78664-2109
Lauren Burns	502 Windsor Rd, Round Rock TX 78664-2109

11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities. If none, check box: ☐ None

Full Name	Complete Mailing Address

12. Tax Status (For completion by nonprofit organizations authorized to mail at special rates) (Check one)
The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes:
☐ Has Not Changed During Preceding 12 Months
☐ Has Changed During Preceding 12 Months (Publisher must submit explanation of change with this statement)

PS Form 3526, September 1998 — See Instructions on Reverse

13. Publication Title: MICROpendium	14. Issue Date for Circulation Data Below: September/October 1996
15. Extent and Nature of Circulation	Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue Published Nearest to Filing Date
a. Total Number of Copies (Net press run)	1087	900
b. Paid and/or Requested Circulation (1) Sales Through Dealers and Carriers, Street Vendors, and Counter Sales (Not mailed)	145	155
(2) Paid or Requested Mail Subscriptions (Include advertiser's proof copies and exchange copies)	600	530
c. Total Paid and/or Requested Circulation (Sum of 15b (1) and (2))	750	669
d. Free Distribution by Mail (Samples, complimentary, and other free)	93	86
e. Free Distribution Outside the Mail (Carriers or other means)		
f. Total Free Distribution (Sum of 15d and 15e)	93	86
g. Total Distribution (Sum of 15c and 15f)	843	755
h. Copies not Distributed (1) Office Use, Leftovers, Spoiled	244	145
(2) Returns from News Agents		
i. Total (Sum of 15g, 15h(1), and 15h(2))	1087	900
Percent Paid and/or Requested Circulation (15c / 15g x 100)	90.56	88.6

16. Publication of Statement of Ownership: ☐ Publication required. Will be printed in the November/December 1996 issue of this publication. ☐ Publication not required.

17. Signature and Title of Editor, Publisher, Business Manager, or Owner: Lauren Burns Editor — Date: 10/15/96

I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanctions (including fines and imprisonment) and/or civil sanctions (including multiple damages and civil penalties).

Instructions to Publishers

1. Complete and file one copy of this form with your postmaster annually on or before October 1. Keep a copy of the completed form for your records.
2. In cases where the stockholder or security holder is a trustee, include in items 10 and 11 the name of the person or corporation for whom the trustee is acting. Also include the names and addresses of individuals who are stockholders who own or hold 1 percent or more of the total amount of bonds, mortgages, or other securities of the publishing corporation. In item 11, if none, check the box. Use blank sheets if more space is required.
3. Be sure to furnish all circulation information called for in item 15. Free circulation must be shown in items 15d, e, and f.
4. If the publication had second-class authorization as a general or requester publication, this Statement of Ownership, Management, and Circulation must be published; it must be printed in any issue in October or, if the publication is not published during October, the first issue printed after October.
5. In item 16, indicate the date of the issue in which this Statement of Ownership will be published.
6. Item 17 must be signed.

Failure to file or publish a statement of ownership may lead to suspension of second-class authorization.

PS Form 3526, September 1998 (Reverse)

GRIDLUCK —

(Continued from Page 24)

```
 THEN WP(VP,HP)=1 :: CALL HC
HAR(X-1,Y,36,2):: GOTO 39 EL
SE WP(VP,HP)=0 :: GOTO 38
38 NEXT C
39 NEXT Y :: HP=0 :: NEXT X
:: VP=0 :: WN=0
40 ! ~~~~~ CHECK FOR WIN BY
    FOUR CORNERS MATCHED
41 IF WP(1,1)=1 AND WP(1,G)=
1 AND WP(G,1)=1 AND WP(G,G)=
1 THEN WN=10 :: GOTO 60 ELSE
 WN=0
42 ! ~~~~~ CHECK FOR WIN BY
    4 IN A SQUARE MATCHED
43 CALL CW(WP(,),G-1,1,G-1,5
,WN,2):: IF WN>0 THEN 60 ELS
E WN=0
44 ! ~~~~~ CHECK FOR WIN BY
    4 IN A ROW VERTICAL
45 CALL CW(WP(,),G-3,1,G,1,W
N,3):: IF WN>0 THEN 60 ELSE
WN=0
46 ! ~~~~~ CHECK FOR WIN BY
    4 IN A ROW HORIZONTAL
47 CALL CW(WP(,),G,1,G-3,2,W
N,3):: IF WN>0 THEN 60 ELSE
WN=0
48 ! ~~~~~ CHECK FOR WIN BY
    4 IN ROW DIAGONAL
49 CALL CW(WP(,),G-3,1,G-3,3
,WN,5):: IF WN>0 THEN 60 ELS
E WN=0
50 CALL CW(WP(,),G-1,3,G,4,W
N,5):: IF WN>0 THEN 60 ELSE
WN=0
51 ! ~~~~~ NO WIN PLAY AGAIN
?
52 DISPLAY AT(24,1):"NO WIN
THIS TIME! REDO? Y/N" :: CAL
L KEY(3,K,S):: IF S<1 THEN 5
2 ELSE IF K=89 THEN 58
53 PRINT "**BE READY TO WIN
NEXT TIME**":"**ARE YOU REAL
LY READY?**" :: STOP
55 DISPLAY AT(5,1)ERASE ALL:
"$$$$$$$ GRIDLUCK $$$$$$$" :
: BR=100 :: CALL MAGNIFY(2):
: CALL SPRITE(#1,32,16,24,24
0,0,-10)
56 DISPLAY AT(7,1):"4 FOR 4X
4 GRID":"5 FOR 5X5 GRID":"6
FOR 6X6 GRID":"7 FOR 7X7 GRI
D":"8 FOR 8X8 GRID"
57 DISPLAY AT(12,1):"OTHER N
UMBERS TO RANDOMIZE" :: INPU
T "YOUR CHOICE? ":N
58 CALL CLEAR :: IF N<4 AND
N>8 THEN 12
59 DISPLAY AT(23,1):"YOU NOW
 HAVE $";BR :: INPUT "HOW MU
CH TO WAGER? ":B :: BR=BR-B
:: IF BR<0 THEN PRINT "YOUR
BROKE!" :: GOTO 53 ELSE CALL
 CLEAR :: GOTO 12
60 ! ~~~~~ ANNOUNCE WIN
61 CALL SCREEN(14):: DISPLAY
 AT(1,1):"YOU WON $";B*WN*6
:: BR=BR+B*WN*G :: WN=0
62 INPUT "PLAY AGAIN? Y/N ":
Z$ :: IF Z$="Y" THEN 59
63 GOTO 53
64 ! **** TESTING PROGRAM
  STOP PROGRAM AT PROMPT FOR
  NUMBERS TO PICK AND TYPE
  ONE OF THE FOLLOWING LINES
65 ! TO CHECK WP() ARRAY >>>
 FOR X=1 TO 8 :: FOR Y=1 TO
8 :: PRINT WP(X,Y);:: NEXT Y
 :: PRINT "" :: NEXT X
66 ! TO CHECK D$() ARRAY >>>
 FOR X=1 TO N^2 STEP G :: FO
R Y=0 TO G-1 :: PRINT D$(X+Y
);:: NEXT Y :: PRINT "" :: N
EXT X
67 ! TO WIN EVERY TIME >>>>
 CHANGE LINE 16 FROM X$=STR$
(X) TO X$=STR$(J) THEN PICK
CONSECUTIVE NUMBERS OR NUMBE
RS THAT ARE THE VALUE OF G
68 ! OR G+1 OR G-1 MORE THAN
 THE PREVIOUS NUMBER OR PICK
 THE CORNERS WITH 1,G,G*G-(G
-1),G*G
69 ! ON A 8X8 GRID THAT WOUL
D BE 1,8,57,64 OR ON A 4X4 G
RID THAT WOULD BE 1,4,13,16
OR ON A 6X6 GRID THAT WOULD
BE 1,6,31,36
70 ! THE ODDS AGAINST WINNIN
G ON GRIDLUCK ARE BIGGER ON
THE LARGER GRIDS BUT THE AMO
UNT OF A WIN IS ALSO BIGGER.
71 ! THE RANDOM PICK BY THE
COMPUTER IN LINES 13 & 16 MA
KE IT HARD TO WIN.
72 ! YOU CAN WIN BY PICKING
THE 4 CORNER NUMBERS OR BY P
ICKING ANY 4 CONSECUTIVE NUM
BERS IN A ROW HORIZONTALLY V
ERTICALLY OR DIAGONALY.
73 ! OR BY PICKING 4 NUMBERS
 IN A SQUARE ADJOINING .
 HAVE FUN WITH IT!!!!!!
100 ! #### SUBPROGRAM ####
101 SUB CW(WP(,),R,C,D,T,WN,
M)
102 FOR X=1 TO R :: FOR Y=C
TO D :: W=0 :: IF T=5 THEN 1
16
103 FOR Z=0 TO 3 :: ON T GOT
O 105,108,111,114
104 NEXT Z :: IF W=4 THEN WN
=M :: GOTO 125 ELSE W=0 :: G
OTO 125
105 IF WP(X+Z,Y)=1 THEN W=W+
1 ELSE W=0 :: GOTO 122
106 GOTO 104
108 IF WP(X,Y+Z)=1 THEN W=W+
1 ELSE W=0 :: GOTO 122
109 GOTO 104
111 IF WP(X+Z,Y+Z)=1 THEN W=
W+1 ELSE W=0 :: GOTO 122
114 IF WP(X+Z,Y-Z)=1 THEN W=
W+1 ELSE W=0 :: GOTO 122
115 GOTO 104
116 IF WP(X,Y)=1 AND WP(X,Y+
1)=1 AND WP(X+1,Y)=1 AND WP(
X+1,Y+1)=1 THEN WN=M :: GOTO
 125 ELSE 122
122 NEXT Y :: NEXT X
125 SUBEND
```

Nominations sought for Peterson award

Jim W. Krych recently announced on the comp.sys.ti forum that nominations are open for the 1997 Jim Peterson Achievment Award. Awards are given in the areas of Software, Hardware, Community Achievment and 9640 Achievment (can be hardware OR software for the 9640).

Nominations may be sent to ab453@cleveland.freenet.edu.

MagicFile Manipulator

XBASIC program combines file transfer capabilities with file management functions

By: BEN HATHEWAY

This program when run on a TI99/4A, allows the rapid transfer of files between the computer on which it is running (the host) and another computer directly connected to it with an RS232 cable. In addition, file management features allow the protecting, unprotecting, renaming and deletion of files on the host system. Also, a catalog feature is provided and D/V80 and D/F128 text files may be viewed with or without CR/LF's inserted.

Baud rates are selectable from 300 to 19200 baud. The host computer must be running a terminal emulator program designed for the computer that supports XMODEM file transfer protocol.

Version 3 features support for floppies and hard drives connected to the Myarc HFDC. Full pathnames are used to catalog directories and subdirectories and transfer files to and from hard drive directories and floppy disks. File transfers to and from subdirectories on floppies are not supported.

This program when run on a TI99/4A, allows the rapid transfer of files between the computer on which it is running (thehost) and another computer directly connected to it with an RS232 cable.

Version 3 also allows you to select the RS232 port of the host so that the program may be more compatible with your system configuration.

Cable connections from the host to the remote computer may be made with as few as three wires. For Ports 1 and 3, pins 2, 3, and 7 are used. If connecting to an PC, pins 2 and 3 on each end are connected straight through to pins 2 and 3, respectively. If the remote computer is another TI or Geneve, pins 2 and 3 are cross connected. (Of course, Port 3 must be on a second RS232 card specially modified to serve as ports 3 and 4.) If you are using a "Y" cable on the RS232, then the same pin numbers would be used for ports 2 and 4 on the end of the "Y" cable designated for those ports.

USING MAGICFM

First, run the appropriate terminal program on the remote computer. If the remote computer is capable of reliable operation at 19200 baud, then the use of this

(See Page 27)

1996 TI FAIRS

MARCH

1996 TI Workshop, TI99/4A User Group U.K., March 16, Wheatsheaf Public House, Sandbach, Cheshire, England. Contact Trevor Stevens, chairman, 249 Southwell Rd. East, Rainworth, Notts, NG21 0BN, UK, or call the MOBB BBS at 01623 491282.

Dutch TI Users Group Annual Meeting, March 23, Buusthuis Kremerstraat 241 Utrecht, The Netherlands. Contact Berry Harmsen, chairman, 1E Oosterparkstraat 141E, 1091 GZ, Amsterdam, The Netherlands, (phone) (31) 20-6941047.

NOVEMBER

11th International TI99/4A and Geneve Computer-Treffen, Nov. 1-3, Freizeithem Vorsfelde, Am Sportplatz 5, D38448 Wolfsburg, Germany. Contact Martin Zeddies, Haupstr. 28, D-38448 Wolfsburg-Reislingen, Germany, Phone/fax number +Germany-5363-71125.

14th Chicago TI International World Faire, Nov. 9, Evanston Public Library in Evanston, Illinois. Contact Hal Shanafield, Faire chairman, at (847) 864-8644, or Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009

1997 TI FAIRS

APRIL

Fest West '97, April 5, San Jose Civic Auditorium, San Jose, California. Contact Fest West '97 c/o Don O'Neil, 3297 Woody Lane, San Jose, CA 95132, or call (408) 934-0352.

MAY

Multi Users Group Conference, May 23-24, Ohio State University, Lima Campus. Contact Charles Good, P.O. 1ox 447, Venedocia, OH 45894. Phone (419) 667-3131. Preferred e-mail address good.6@osu.edu.

This TI event listing is a permanent feature of MICROpendium. User groups and others planning events for TI/Geneve users may send information for inclusion in this standing column. Send information to MICROpendium Fairs, P.O. Box 1343, Round Rock, TX 78680.

MAGICFM —

(Continued from Page 26)

speed will provide far superior performance.

Boot Magic FM on the host. Follow the prompts and select the desired baud rate and RS232 port. Then the screen will switch to 40-column mode and all commands from that point on may be entered from the remote computer. There is no need to move back and forth between keyboards.

Most of the prompts are self-explanatory. When entering pathnames, the period at the end may be left off. The program will add it in case it is omitted. Do not include the filename when it asks for a path. For floppies, enter DSKx where x is a valid drive in your system. You may view a D/V80 or D/F128 file with or without CR/LF's inserted. In addition, a catalog may be paused by pressing P, or CTRL S. Resuming is with enter, R, or CTRL Q. Aborting may be done with A or S or CTRL X.

TRANSFERRING FILES

When transferring files, as with all operations, be sure to enter a valid pathname or the desired function will not be performed. When transferring a file from the host or when viewing a file, the presence of the desired file is checked for validity and appropriate file type before the operation is performed. If the file to be transferred from the host is a D/F128 unprotected file, which may be a non-TI file, you are given the option of adding the TIFILES header onto the file, or not as you wish.

When transferring a file to the host, the desired path is checked for the presence of a file with the same name and, if found will notify you of that fact.

Some limited modem support is provided in the form of a modem initialization string in line 115 in the MAGICFM file. This string turns off the modem echo, the modem result codes, and sets the Hayes-compatible modem to answer on the first ring. To use the program with a modem, set it up with the baud rate which the caller will use. He must use the predetermined baud rate. Upon answering the line, the caller can press enter to re-display the menu. Be sure not to quit the program if you wish to call back later. Instead, just hang up from the remote location. Line 115 must be un-commented to use this feature.

A test mode is provided. Press function 7 on the host to turn on and function 9 on the host or CTRL O on the remote computer.

Please note that when renaming or deleting files, that the file to be acted upon will first be unprotected and in the case of renaming, the new file will remain unprotected.

Also, note that when transferring files from the Host with Xmodem, the HFDCC "file-backed up" bit is stripped so as to be compatible with receiving systems. Files received by the host are taken as-is.

It is hoped that this program will fill a need where users have two computer types and wish to allow them to communicate together.

LOAD

```
0 !************************ !0
31
1 !*                       * !0
87
2 !*    Generated By       * !0
65
3 !*    SysTex  V1.0       * !1
08
4 !*      (C) 1985         * !2
26
5 !*   By Barry Boone      * !1
33
6 !*                       * !0
87
7 !************************ !0
31
8 CALL INIT :: CALL LOAD(819
6,254,0)!126
9 CALL LINK("SLOAD")!067
10 CALL LINK("SETUP")!097
20 RUN "DSK1.MAGICFM" !177
```

MAGICFM

```
1 GOTO 100 :: B$,D$,ERR$,FB$
,FI$,FQ$,P$,PATH$,PROG$ :: C
,D,DD,DR,DX,DY,F,I,K,MXT,MYT
,PORT,PROT,PU,PY,SP,STL,V,VA
R,X :: CALL CLEAR :: CALL LI
NK :: CALL LOAD :: !@P-
100 ON ERROR 880
110 CALL CLEAR :: PRINT :"1)
 300":"2) 1200":"3) 2400":"4
) 4800":"5) 9600":"6) 19200"
: :
111 INPUT "Please select des
ired baud  rate: ":SP :: IF
SP<1 OR SP>6 THEN 110
112 CALL CLEAR :: INPUT "Ple
ase select desired RS232 por
t (1-4) ":PORT :: IF PORT<1
OR PORT>4 THEN 112
115 OPEN #1:"RS232/"&STR$(PO
RT)&".BA=300.DA=8.PA=N" :: P
RINT #1:"AT E0 Q0 S0=1 S7=30
" :: CLOSE #1
120 ON ERROR STOP :: CALL LO
AD(-24576,40,0,0,0)
130 CALL LINK("FCOL"):: CALL
 CLEAR :: CALL LINK("P"&STR$
(PORT)):: CALL LINK("S"&STR$
(SP))
140 CALL LINK("PRINTC",41,RP
T$("-",38)):: CALL LINK("PRI
NTC",17,"HOST")
150 CALL LINK("PRINTC",841,R
PT$("-",38))
160 CALL LINK("PRINTC",884,"
  Magic File Manipulator V3.
0"):: CALL LINK("PRINTC",931
,"By Ben Hatheway")
170 P$="`~ Magic File Manipu
lator~~  By Ben Hatheway~~
with support for the HFDCC"
:: GOSUB 530
180 P$="~~ F)ile Management~
 V)iew DV/80 or DF/128 file~
 X)modem transfers~ Q)uit~~
  Select: "
190 GOSUB 530 :: GOSUB 570 :
: IF B$="" THEN 170
200 IF V=81 THEN 300 ELSE IF
 V=70 THEN 740 ELSE IF V=88
THEN 310 ELSE IF V=86 THEN 2
10 ELSE 180
210 P$="~~Host Path: " :: GO
SUB 530 :: STL=39 :: GOSUB 5
60 :: PATH$=B$ :: IF PATH$="
" THEN 180 ELSE IF SEG$(PATH
$,LEN(PATH$),1)<>"." THEN PA
TH$=PATH$&"."
```

(See Page 28)

MAGICFM —

(Continued from Page 27)

```
220 P$="~~Filename: " :: GOS
UB 530 :: STL=10 :: GOSUB 56
0 :: IF B$="" THEN 180 ELSE
FI$=PATH$&B$
230 CALL LINK("DIR",FI$,DD,D
X):: IF DD>0 THEN GOSUB 890
:: GOTO 210 ELSE D$=FI$
240 IF DD=0 THEN 280
250 P$="~~Do you wish carria
ge returns and~linefeeds (no
rmal print)? Y/n: " :: GOSUB
 530 :: GOSUB 570 :: IF V=78
 THEN PROT=1 :: CALL LINK("P
RINTX","`"):: GOTO 270
260 GOSUB 290
270 IF DD=-1 THEN 600 ELSE I
F DD<-1 THEN 610
280 P$="~~File must be DV/80
 or DF/128~~ Press enter." :
: GOSUB 530 :: GOSUB 570 ::
GOTO 170
290 P$="` P or Ctl S to paus
e~ R or Ctl Q to resume~ A o
r S to abort~~" :: GOSUB 530
 :: RETURN
300 P$="`~Thank you for usin
g~Magic File Manipulator." :
: GOSUB 530 :: END
310 !FILES
320 P$="`~~  U)pload (to hos
t)~  D)ownload (from host)~
 Q)uit~~ Select=>" :: GOSUB
530 :: GOSUB 570
330 IF B$="" THEN 320
340 IF V=68 THEN 460
350 IF V=81 THEN 170
360 IF V<>85 THEN 320
370 P$="~~Upload to Host~Pat
h: " :: GOSUB 530 :: STL=39
:: GOSUB 560 :: PATH$=B$ ::
IF PATH$="" THEN 320 ELSE IF
 SEG$(PATH$,LEN(PATH$),1)<>"
." THEN PATH$=PATH$&"."
390 P$="~Filename: " :: GOSU
B 530 :: STL=10 :: GOSUB 560
 :: IF B$="" THEN 320 ELSE F
I$=PATH$&B$
400 CALL LINK("DIR",FI$,DD,D
X):: IF DD<=0 THEN P$="~~Fil
ename already used.." :: GOS
UB 530 :: GOTO 370
410 MXT=0 :: C=1 :: PROG$=FI
$ :: P$="~Start upload on re
mote computer.." :: GOSUB 53
0 :: FOR X=1 TO 100 :: NEXT
X
420 GOSUB 430 :: GOTO 310
430 D=0 :: CALL LINK("XMODEM
",PROG$,C,D,MXT):: CALL LINK
("SETUP")
440 DD=C :: GOSUB 890 :: RET
URN
450 !
460 P$="~~Download from Host
~Path: " :: GOSUB 530 :: STL
=39 :: GOSUB 560 :: PATH$=B$
 :: IF PATH$="" THEN 310 ELS
E IF SEG$(PATH$,LEN(PATH$),1
)<>"." THEN PATH$=PATH$&"."
470 C=0 :: P$="~Filename: "
:: GOSUB 530 :: STL=10 :: GO
SUB 560 :: IF B$="" THEN 310
 ELSE PROG$=PATH$&B$
480 CALL LINK("DIR",PROG$,DD
,DX):: IF DD>0 THEN P$="~~Fi
le "&B$&" not found" :: GOSU
B 530 :: GOTO 460
490 IF DD<>-2 THEN MXT=1 ::
GOTO 510
500 P$="~~This is a DF/128 f
ile. Do you wish a~TIFILES h
eader? (Normal)Y/n: " :: GOS
UB 530 :: GOSUB 570 :: IF V=
78 THEN MXT=0 ELSE MXT=1
510 P$="~Start download on r
emote computer." :: GOSUB 53
0 :: GOSUB 430
520 GOTO 310
530 IF PROT=0 THEN CALL LINK
("PRINT",P$):: F=1 :: GOTO 6
70
540 CALL LINK("PRINTG",P$)::
 F=1 :: GOTO 670
550 CALL LINK("PRINTX",P$)::
 F=1 :: GOTO 670
560 CALL LOAD(-24576,STL)::
CALL LINK("IN",B$):: F=1 ::
GOTO 670
570 CALL LINK("ONE",B$):: F=
4 :: GOTO 670
580 CALL LINK("NUMIN",B$)::
F=3 :: GOTO 670
590 I=0 :: IF PROT=1 THEN 32
767
600 OPEN #4:D$,INPUT :: GOTO
 620
610 OPEN #4:D$,INPUT ,DISPLA
Y ,FIXED 128
620 IF EOF(4)THEN 660
630 LINPUT #4:P$ :: I=I+1
640 IF PROT=0 THEN P$="~"&P$
 :: GOSUB 530 :: IF P$="S" T
HEN 660 ELSE 620
650 GOSUB 540 :: IF P$="S" T
HEN 660 ELSE 620
660 PROT=0 :: CLOSE #4 :: P$
="~~Press Enter" :: GOSUB 53
0 :: GOSUB 570 :: GOTO 170
670 !
680 ON F GOTO 690,700,700,71
0
690 RETURN
700 IF B$="" THEN VAR,V=0 EL
SE VAR,V=VAL(B$):: GOTO 3276
7
710 IF B$="" THEN VAR,V=0 ::
 RETURN ELSE IF ASC(B$)>96 T
HEN VAR,V=ASC(B$)-32 :: B$=C
HR$(VAR):: RETURN ELSE VAR,V
=ASC(B$):: RETURN
720 IF K=96 THEN CALL LINK("
FCOL")
730 RETURN
740 P$="~~C)at path~P)rotect
~R)ename~U)nprotect~D)elete~
Q)uit: " :: GOSUB 530 :: GOS
UB 570 :: IF V=67 THEN 840 E
LSE IF V=80 THEN PU=1 ELSE I
F V=85 THEN PU=0 ELSE IF V=8
2 THEN PU=2 ELSE IF V=68 THE
N PU=3 ELSE IF V=81 THEN 170
 ELSE 740
750 P$="~~Host Path: " :: GO
SUB 530 :: STL=39 :: GOSUB 5
60 :: PATH$=B$ :: IF PATH$="
" THEN 740 ELSE IF SEG$(PATH
$,LEN(PATH$),1)<>"." THEN PA
TH$=PATH$&"."
760 P$="~~Filename: " :: GOS
UB 530 :: STL=10 :: GOSUB 56
0 :: IF B$="" THEN 740 ELSE
FI$=PATH$&B$ :: FB$=B$
770 CALL LINK("DIR",FI$,DD,D
X):: IF DD<0 OR DD=0 THEN 78
0 ELSE GOSUB 890 :: GOTO 740
780 IF PU=2 THEN 800 ELSE IF
 PU=3 THEN 820
790 IF PU=0 THEN CALL LINK("
```

(See Page 29)

MAGICFM —

(See Page 28)

```
UNPROT",FI$):: IF PY=1 THEN
PY=0 :: RETURN ELSE 740
795 IF PU=1 THEN CALL LINK("
PROT",FI$):: GOTO 740
800 P$="~~New Filename: " ::
 GOSUB 530 :: GOSUB 560 :: I
F B$="" THEN 740 ELSE FQ$=B$
 :: PY=1 :: PU=0 :: GOSUB 79
0
810 CALL LINK("RENAME",PATH$
&FQ$,DD,FB$):: GOSUB 890 ::
GOTO 740
820 P$="    Delete? y/N " ::
 GOSUB 530 :: GOSUB 570 :: I
F B$<>"Y" THEN 740
830 PU=0 :: PY=1 :: GOSUB 79
0 :: DELETE FI$ :: GOTO 740
840 ! Disk Catalog
850 P$="~~Host Path to catal
og:~ " :: GOSUB 530 :: STL=3
9 :: GOSUB 560 :: PATH$=B$ :
: IF PATH$="" THEN 740 ELSE
IF SEG$(PATH$,LEN(PATH$),1)<
>"." THEN PATH$=PATH$&"."
860 GOSUB 290 :: CALL LINK("
CAT",PATH$,X)
870 P$="~~  Press Enter" ::
GOSUB 530 :: GOSUB 570 :: GO
TO 740
880 ON ERROR 880 :: RUN
890 IF DD=0 THEN P$="~~Succe
ssful."
891 IF DD=1 THEN P$="~~File
not found."
892 IF DD=2 THEN P$="~~Canno
t open file.~Check Path/file
name."
893 IF DD=3 THEN P$="~~Too m
any retries, aborted."
894 IF DD=4 THEN P$="~~Trans
fer cancelled by Remote."
895 IF DD=5 THEN P$="~~Trans
fer cancelled by Host."
896 IF DD=6 THEN P$="~~Inval
id Pathname."
897 IF DD=7 THEN P$="~~Error
 in renaming file."
898 P$=P$&"~~  Press <enter>
 " :: GOSUB 530 :: GOSUB 570
 :: RETURN
1000 !@P+
1010 RUN
```

CorComp cures

Suggestions offered to correct problems with CorComp cards

This comes from the June 1996 issue of the VAST News, *newsletter of the Valley of the Sun TI99ers in Phoenix, Arizona.)—Ed.*

The following concerns problems and solutions involving various cards manufactured by CorComp. Any modifications that readers pursue based on information in this article is the responsibility entirely of the reader.

Triple-Tech Card: The main problem here has been due to the lithium battery exploding. Somehow current is being allowed to reach the battery from the P-Box. Lithium batteries were never designed to be recharged and they do not seem to appreciate you attempting to do so. The cure is to remove a 100-ohm resistor R7 and replace it with a diode. The anode side of the diode must face toward the battery holder.

The main problem with the Triple-Tech card has been due to the lithium battery exploding. Somehow current is being allowed to reach the battery from the P-Box.

PEB RS-232 Card and MES Revisions: Pin 38 of the TMS9901 IC should have a 1.5K ohm resistor to ground. If it does not have this resistor, it could cause you to have no menu screen when using the CorComp Floppy Disk Controller (FDC). DTR/1 and DTR/2 inputs on the RS-232 port need a 3.3K ohm resistor to +5 volts (pull-up).

PEB Floppy Disk Controller and MES: Pin 15 of the 74LS123 should have a 200K ohm resistor to +5V. Possible problems without this resistor are 1) No Diskette or Drive Error, 2) problems when copying with one drive, 3) cannot initialize a brand new disk, 4) will not run Miller's Graphics Advanced Diagnostics program. After you add the resistor and if the drive motor goes off and on repeatedly, then the 74LS123 is shot. Replace it. Any 2793A FDC ICs, unless having an "07" number marked prominently on them, are bad. Usual failure mode is failure during reading or writing in double density mode. If the 2793 has no "A" behind it, it is OK. If you have a 2793A, replace it.

Cards with Voltage Regulator IC's and very much the FDC: Heat sinks are required on these regulators. Silicone grease thermal compound is highly recommended. Without the heat sink you could have numerous problems, as the regulator can and will go into thermal shutdown, which limits current tot the card.

FDC alignment for the cards using the 2793 FDC IC: Allow computer and card to warm up a minimum of 15 minutes. Install a jumper clip across the W5 connector. Adjust variable capacitor so that frequency reads 250K Hz. Use an oscilloscope and adjust R6 (10K) to .3 uS. Use the oscilloscope again and adjust (50K) to .5 uS.

User-defined functions and subprograms in Extended BASIC

By W. THOMAS BOUSSUM

This article comes from the October/November/December 1996 issue of A Byte of Info, *newsletter of the Reading-Berks 99ers in Pennsylvania.—Ed.*

I believe that it was Julius Caesar who originally stated "Divide and conquer." This is certainly true as a military reality, but it is also true in writing computer programs. This reality is the real inspiration for this article. A feature of TI Extended BASIC that is often overlooked by users of the TI99/4A is the use of "user-defined" functions and subprograms. As many of you already know, I use my computers for some rather specialized types of programs.

Since the types of program that I write are specialized, especially in the areas of mathematics, physics, astronomy and astrophysics, I often need to write some very specific routines to fit the need that I have. The use of user defined functions and subprograms allows me to "divide and conquer." All forms of BASIC have this capability, to some extent, and many programmers simply overlook these shortcuts that can save both computing time and memory.

TI Extended BASIC allows us to save a file as a "merged" file. Thus, we can write a routine once that will be used in many programs that will be written subsequently.

NOTE: *The "merged" save will work only with disk systems. It cannot be used with a cassette system, since there is no way that the cassette system can find a named file.*

To save a file in merged format, the user simply enters SAVEDSK*d*.FILENAME,MERGE, where *d* is the drive identification (1, 2, 3, etc.) and FILENAME is the name of the program as it should appear on the disk on which it is saved. Then when he enters a program that needs the routine, he simply creates a hole in the program and enters "MERGE DSK*d*FILENAME"; There is a note of caution here, however. *Do not* use line numbers in the program that you are entering to conflict with those of the merged routine. If you do, the merged program will overwrite the lines that carry the same numbers.

The advice that I would give to programmers who plan to use merged files is to build up a library of routines, numbered 20000-32767, for example, and use lines numbers below 20000 for the developing programs that will use the merged files.

The advice that I would give to programmers who plan to use merged files is to build up a library of routines, numbered 20000-32767, for example, and use lines numbers below 20000 for the developing programs that will use the merged files. No matter what numbering scheme is used the user would be advised to print hard copies of any files that he plans to save a s a merged file. Then he should put them into a reference notebook and refer to them whenever he writes a program that will use these merged files.

Merged programs might use *user-defined* functions or subprograms that are referred to or CALLed from the main program. A function in a computer programming language relates to the definition of the word "function" in mathematics. This probably goes back to the early days of computing, when the first machines were used primarily as *number crunchers*. In mathematics, we say, for example, that *y* is a function of *x* to state that there is a rule that defines a relationship between *x* and *y*, denoted as *y=f(x)*. When the rule is imposed in mathematics we are able to find at least one value of *y* for each value of *x* that we assign to the *variable x*.. As an example of a mathematical function, consider the equation *y=x+1*. The functional rule states that for any value of *x* we first square *x* then add 1 to the new value. Thus, if we assign a value of 2 to the variable *x* then when we square 2 we obtain 4. Adding 1 to 4 gives us a value of 5 for *y*.

Predefined functions in TI Extended BASIC do exactly the same thing. The predefined functions are listed on page 21 of the TI Extended BASIC manual. Many of these functions are mathematics functions (SIN, COS, LOG, ATN, EXP, etc.). However, there are some that are strictly computer related. Among these are CHRS, EOF LEN, POS, REC, RPTS, SEGS, STRS and TAB. There is one common element to all of these functions; that is that they use at least one *parameter* enclosed in parentheses. That parameter is also known as the *variable*.

For example, if we wanted to have the computer print 10 character "A"s on the screen, we could accomplish that by typing RPT$("A",10). The computer would respond by printing the characters AAAAAAAAAA on the monitor screen. As you can see, this function does exactly what PRINT "AAAAAAAAAA" would do if we typed the command longhand. I leave it to the user to read the TI Extended BASIC manual to find out the use of the other predefined functions. These functions are very useful in Extended BASIC programming and should not be neglected by anyone who wants to use his TI as a programming tool.

In addition to the predefined programs, there is a class of instructions known as "user-defined functions." Using these operations is as easy, or difficult, as deciding what you want to do, then define a func-

(See Page 31)

SUBPROGRAMS —

(Continued from Page 30)

tion to do it. As with the predefined functions, a parameter is required to serve as the variable for the user-defined operation.

A simple mathematical function is defined below. It is based upon the example that I gave above of $y=x^2+1$.

```
...     ...     ...
200 DEF Y(X)=X^2+1
...     ...     ...
5000 FOR I=1 TO 20
5010 PRINT X,Y(X)
5020 NEXT I
...     ...     ...
```

Notice that I defined *y* at line number 200. The reason I did not use a large line number was that when a "function definition" is given, the definition *must* appear before the function is used. So with function definitions it is best to place them as close to the beginning of a program as is possible. If we were to refer to the function in line 120 it would flag an error.

There is another "user-defined" operation that is permissible with TI Extended BASIC, the *user-defined subprogram.* With this technique it is possible to define a program segment so that it may be CALLed from a running program. There are several built-in subprograms that are available to TI Extended BASIC. Perhaps the most widely used is CALL CLEAR.

When CALL CLEAR is invoked, the computer writes the blank space character (ASCII 32) to every screen position on the monitor. This subprogram is handled on the EPROM (Erasable Programmable Read Only Memory) in the Extended BASIC cartridge, along with the other built-in subprograms. For a full definition of the built-in subprograms see the TI Extended BASIC manual.

Beyond the built-in subprograms again TI Extended BASIC has the capability for permitting the user to define his own subprograms to be called in a running program.

A subprogram that I use often in programming is the mathematical operation of taking a factorial of a number. *n! (n factorial)* is simply the product of every number from 1 to *n*. *0!* is defined as being equal to 1. A function which will provide *n!* for values of *n* which are less than or equal to 84 follows:

```
20000                                   SUB
FACTORIAL(N,NFAC(N))
20010 NFAC(0)=1
20020 FOR I=1 to N
20030 NFAC(I)=I*NFAC(I-1)
20040 NEXT I
20050 SUBEND
```

This subprogram acts like a subroutine, with the SUBEND of line 20050 acting like a RETURN at the end of a subroutine. When CALL FACTORIAL(X1,X2) is invoked in the running program, the operation branches to the FACTORIAL subprogram and executes to calculate the value of NFAC. Parameter passing is handled through the variables X1 and X2 in the main program. These numbers are then passed to the values N and NFAC(N) in the subprogram. After the calculation has been performed the new values are returned to the main program as X1 and X2, which may be used for further calculations.

Try these routines or use them for examples in your programming. I think that you will find they will save you both time and memory.

Book has suggestions to deal with computer pain

Dr. Sheik Imrhan has written a new book, *Help! My Computer Is Killing Me: Preventing Aches and Pains in the Computer Workplace.*

Advice he gives in the book includes:

• Don't put your monitor on top of your hard drive (this can cause neck aches); set it on your desktop. You should be looking slightly down at your monitor instead of slightly up.

• If your feet do not rest flat on the floor, get a footrest to prevent the strain put on your legs when they don't reach the ground.

• To avoid eyestrain, look away occasionally from your computer screen to distant objects. This simple exercise uses different muscles in the eye, allowing those used to look at close objects (such as your screen) to rest for a moment.

• "Microbreaks," which last less than a minute but can be taken often, can be more effective than traditional 10- or 15-minute breaks taken only a few times a day. A microbreak can be as simple as stretching your legs or flexing your wrists.

• The least stressful posture, the one that imposes minimum overall stress in your muscles and joints, closely resembles the horseback rider's posture without the wide splaying of the legs.

• The sharper you bend forward while sitting, the greater the increase in pressure in the spinal discs in your neck and lower back.

• Muscular aches and pains are felt when lactic acid accumulates in muscles. Rubbing and flexing aching muscles often helps prevent accumulation of poisons in them.

• The front edge of the seat of your chair should be curved downward to prevent pressure against the back of you knees.

• A desk that has limited leg space will restrict leg movements and may force you to sit in contorted positions.

• Avoid resting your wrists on the hard edge of the desk or keyboard stand. Try using a pad on the desk edge or a palm rest on the front of the keyboard.

Help! My Computer Is Killing Me sells for $12.95. Publisher is the Taylor Publishing Company, 1550 West Mockingbird Lane, Dallas, TX 75235, (214) 819-8100.

Getting more out of TI-Writer

Editor and formatter tricks you may never have heard of

We're not sure who wrote the following article. We found it in the TIsHUG News Digest. It consists of a series of tips to use the TI-Writer editor and formatter. —Ed.

EDITOR TIPS

• When using replace string, you should turn word wrap off if the document lacks carriage returns or you will end up with one huge paragraph. With word wrap on, replace string will reformat from the amended word to the end of the paragraph.

• This is for use in Funnelweb text editor mode (not TI-Writer) — Want to convert the character case (upper to lower, lower to upper)? Here's how:

Upper to lower — depress CTRL and period.

Lower to upper — depress CTRL and semicolon.

By keeping the keys depressed, the autorepeat function will take effect and every character the cursor passes over will be converted. As long as you continued to hold down the period or semicolon keys, you can release the control key after the first movement of the cursor.

• To save part of a document, insert in front of the filename the first line number you want to save then a space or a comma followed by the last line number you want to save. Example: 5 30 DSK1.FILENAME. This saves lines 5 through 30 on disk drive one in a file called FILENAME.

When using the Find String command, you can specify which column range to search. For example, 5 15 /text/ will look for the string "text" in columns 5 through 15. This feature is also available with Replace String.

• To hide the line numbers on the left side of the screen, press FCTN zero. To display them, press FCTN zero again.

• If you delete a line in error, press CTRL 1, also known as the "oops" key. Your line will be restored. This works only if you haven't pressed any keys after erasing deleting the line.

• When using the search command, remember that the search starts from the position of the cursor. Therefore, to search the complete document, make sure the cursor is located at the beginning of line one before executing the search.

• To backspace beyond the left margin, press CTRL Y. This will temporarily disable the left margin. It will also disable the right margin in Funnelweb. In both cases, the cursor should be next to the margin.

• TI-Writer can save a file in other than the normal D/V80 format by using the PF (PrintFile) command and placing either an "F" or a "C" in front of the filename, F DSK1.FILENAME. The "F" causes the file to be saved in a D/F80 format. The "C" strips control characters from the file as it is saved.

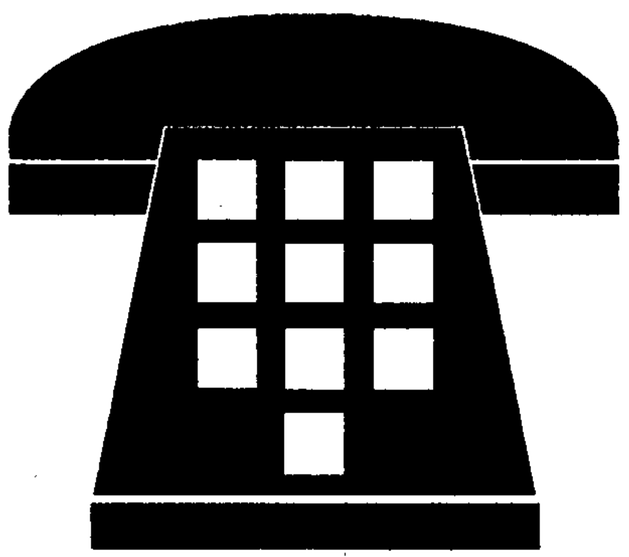
• TI-Writer can be used as a database. Each line must be a record and set up exactly the same. For example, if the data consisted of names, addresses, and phone numbers, then all names must start in the same column. The same goes for the address and phone numbers. There can be no blank lines or lines that contain other kinds of information. You can sort this files using Sort Utility. Once sorted, titles can be added if you are printing it out. Other sort programs can also be used.

(Readers may want to review the May 1989 MICROpendium for an article titled

(See Page 33)

Want to talk to someone at MICROpendium?

You'll need to call between the hours of 9 a.m. and noon Saturdays. If you call at other times, you will probably get an answering machine.



But don't let that bother you. We listen to the answering machine at least once a day and return calls as soon as possible, usually that day.

Call us at 512-255-1512

TI-WRITER —

(Continued from Page 32)

"Program works with TI-Writer to create simple database files." The program accompanying the article is called "Chartbase." — Ed.)

• There are control key equivalents to most function keys, plus a few others. For example, to tab to the right, you can use FCTN 7 or CTRL W. Funnelweb also has CTRL Z. To tab to the left, you can use CTRL T (there is no equivalent function key).

• If you must go to the bottom of a long text file, simply go to the command line and press "S" for ShowLine. At the "enter line number" prompt, type "E" and press Enter. The "E" is a valid line number for the last or end line of a document. This feature is active in all the commands requiring you to enter a line number.

• To avoid a Buffer Full notice, save the file as it gets larger, then use the SD command to see the file size. The buffer is usually becoming full at 92 sectors. Due to the use of run length encoding (RLE) in the buffer area, a full buffer may occur at very different file sizes depending on the nature of the text.

• If you wish to place a carriage return at the end of a line of text — a line without the return will usually occur when you have inserted blank lines in the text and then input text on the lines — place the cursor at the end of the text and press CTRL1 8. This will place a carriage return where you want it and insert a blank line below. If this line is not wanted, you can delete it with FCTN 3.

• You can get a printout of a file with line numbers when printing from the editor by placing the letter "L" and space in front of the printer name in the command line. For example: L PIO. However, this will eliminate the last six characters at the end of each line when using 80-column text. To avoid this, keep your line lengths at 74 or fewer columns.

• You can type anything you want after a carriage return on the same line and it won't print out, but it will save. This is great for text notes that you want available but don't want to appear in printouts.

• When typing a document that uses repetitive phrases or long words you can save time by entering these phrases in short-form or initials, such as TIW for TI-Writer, FTE for Funnelweb text editor, and so on. When you are finished with the document, use the replace string function (RS) to change the words ack to their full spelling. Care must be taken when using this:

The search begins only from the spot that the cursor is on. So, to do the entire document, ensure that the cursor is on the first line before starting.

The search will locate all occurrences of the string. Therefore, if the string

(See Page 34)

TI-WRITER —

(Continued from Page 33)

searched for is "at" it will find word "at" and also the "at" in "cat" and "that." Thus, before having TI-Writer automatically change occurrences of common phrases and syllables, make sure unwanted substitutions won't occur. Otherwise, you can step through each replacement one at a time.

As the text will be reformatted wherever a change is made, it would be wise to review the document to be sure that after execution of the replace function the document remains formatted the way you want it.

• When using the Find String command, you can specify which column range to search. For example, 5 15 /text/ will look for the string "text" in columns 5 through 15. This feature is also available with Replace String.

• If your printer does not have a slashed zero and you want to print it nonetheless, use the following transliteration:

.TL 48:48,8,47

This will cause the normal zero (ASCII 48) to be printed; then backspace (8); and then a printing of a slash (47).

• Two files can be loaded into the editor, assuming the total size is not too large for memory, by loading in the first file then doing a LF and entering E DSKn.YYYY, where "YYYY" is the name of the second file. This will load the second file after the end of the first file.

• If you don't like the windowing when using the 80-column format, then set the margins for zero and 40 and turn off the line numbers (FCTN 0). When you are finished, reset the left and right margins to what you desire and reformat each paragraph.

• You can merge sections of a second file into the current document by using the LF command and then entering the following: 25 7 12 DSK1.YYYY. This will load lines 7 through 12 from file YYYY into the current document starting after line 25.

• If you are using Funnelweb 4.1 or greater, you use the SD command to find out how many lines are in a file. After you have the displayed the directory of a disk on the screen using the SD command, mark the file and then request (V)iew. The line count will be shown at the bottom of the screen as you read through the document.

• If you wish to include a program listing in your document, instead of retyping it into TI-Writer just LIST the program to a disk using in Extended BASIC. For example:

LIST "DSKn.FILENAME"

This saves the program as a D/V80 text file which allows it to be read by TI-Writer. You can load this file into TI-Writer and place a carriage return character at the end of each program line.

(See Page 35)

Tips on using punctuation marks

This is reprinted from the April 1987 issue of Word Play, *the newsletter of the Portland Users of Ninety-Nines.—Ed.*

PUNCTUATION

Typists will always use two spaces after a punctuation mark ending a sentence. TI-Writer, for some strange reason, does things a little differently.

For example:

"." The period (.) — TI-Writer will always put two spaces after every period that has been followed by a single space. This is fine if the period is at the end of the sentence, but what if you are using an abbreviation within a sentence? The formatter will put two spaces here also, but you properly only want one. What you need to do in this case is use the required space symbol (^) after the period of an abbreviation. This will give you the desired one space when using the formatter. (A period followed by no space will appear as just that.)

The exclamation and question marks (!) (?) — In these cases, the formatter will not automatically give two spaces as it properly should. To make your document look correct you will need to add one space and one required symbol (^).

THE PERIOD AND DECIMALS

The formatter thinks that any line which begins with a period is a formatter command and will delete the whole line. If by chance your document contains a value such as .10 and the wraparound caused by Fill and Adjust of the formatter puts it at the beginning of the line, the whole line will disappear. To correct this you could put a zero in front of your decimals (0.10).

ASTERISK AND NUMBERS

If you are printing out of the formatter and your document contains an asterisk followed by two or more numeric digits, the asterisk and the two digits will disappear. For instance, A*256 becomes A6. What's happening here is that the TI-Writer program misinterprets the asterisk and two digits as an instruction to input data from a "value file" as in mail merge. This is described on page 111 of the TI Instruction book. To correct this problem you will need to type two asterisks followed by two dummy numbers, then the actual digits. For example, type A**25256 instead of A*256.

REQUIRES SPACE

If you tie words together for the purpose of underlining (&) or overstriking (@) with the required space (^), the Fill and Adjust of the formatter will leave gaping blanks in your lines. If you tie too many together, the line will extend beyond the right margin. It would be better to put a separate & or @ in front of each word. Be sure to include the spaces between the words. If you want a ^ to appear in your text, you will need to transliterate it; @ and & are typed twice in succession to get them to print.

OTHER PROBLEMS

Other problems have been noted in TI-Writer that cause erratic and destructive commands, but they are not fully documented.

TI-WRITER —

(Continued from Page 34)

FORMATTER

• Outdenting is the reverse of indenting. It is used to start the first line of a paragraph to the left of the remaining text in the paragraph. Here is an example:

.LM 12;IN 7;RM 71

This causes the first line to start at column 7 and subsequent lines to begin at column 12.

• When using the Header or Footer command with the page number, it is possible to have no value printed, such as for the introduction, by using the .PA format command with a value of zero. The page numbering will begin on the following page. A .PA at the end of each page will delay the numbering further.

• If you wish to prevent the form feed at the end of printing when using the formatter, make the last line of your text:

.PL 1

This will suppress the form feed. However, do not forget to reset PL if you have another document to print.

• You can string the formatter commands on the same line, separated with a semicolon. Here is an example: .LM 10;RM 70;IN +5;AD

• If you having problems with formatter commands, make sure they are uppercased.

• When using the .CE (center) command, you must also use the .LM and .RM commands because .CE centers the text between the sete margins, not necessarily the middle of your paper.

• The formatter ensures that you have two spaces after each period. To control this, using the "^" sign in place of the space. For example: Mrs.^E.^Smith.

• If you must have a dot in column one of your text, transliterate it: T L 124:46

This will allow FCTN A to print out a period.

• To create a file with outline feeds, yet formatted, do the following:

Use the formatter to print the text to disk.

Go back to the editor and do a Print File (PF), replacing PIO with C DSKn.FILENAME.

MICROREVIEWS

Speed Reading, P-GRAM Reloader, Assembly Poker, AMS Video Titler, Casino (Klondike) Solitaire

By CHARLES GOOD

SPEED READING
by Bruce Harrison

This excellent assembly language software is not only good for improving your reading speed, it is also useful for proofreading articles you have written on your TI and it can be used as a text file reader. It will display on-screen text files of unlimited size.

When you boot the program either as XB DSK1.LOAD or from EA5 you see a title screen and then are asked for the path of a DV80 file. This prompt will support long path names needed by hard drive users. Next you are asked for the reading speed, between 115 and 1,028 words per minute. Next you are asked for the window size. The answer, a number between 3 and 20, determines the number of text lines displayed simultaneously on the monitor. You are given the option of a 20-, 30- or 40-column display. The lettering in all these displays is the same size no matter what text width you specify, corresponding to what you normally see in the TI's 40-column text mode. Finally you are asked for a "bookmark." This is the record number of the DV80 file where you wish viewing to begin and is useful if you viewed part of the text file in an earlier Speed Reading session and wish to continue where you left off. You can ignore the bookmark prompt by just pressing <enter>. After answering all these prompts you press <enter> to begin speed reading.

The software then reads in a big chunk of text from the designated text file and begins displaying it on screen. You see the number of screen lines designated in your "window" entry as wide as your "column" size entry. The text has word wrap. After a short interval, giving you time to read the entire initial text window, the text starts scrolling up the screen with new material appearing at the bottom of your defined window and old text disappearing out the top. The speed of scrolling is determined by your "words per minute" setting. You can change this speed on the fly, while viewing scrolling text, by pressing the F(aster) or S(lower) key. It is really fun to watch the effect of such an on-the-fly speed change. Your current words per minute speed is indicated at the bottom of the screen. You can pause the scrolling at any time with the space bar. During a pause is a good time to change text scrolling speed. Press <space> again to resume scrolling.

You can exit in the middle of a file by pressing "Q". The program then displays your "bookmark," a number one greater than the last record of the file that you read. When you start Speed Reader again you can input this bookmark number and start reading your text file exactly where you left off.

If you can't remember the instructions or program features, just load the on disk "TESTFILE" into the program and use it to practice your speed reading. TESTFILE is actually the program doc file. Bruce provides an external program to print

(See Page 36)

MICROREVIEWS —

(Continued from Page 35)

these docs if you want a hard copy.

Speed Reading is one of the only 99/4A programs that can display very large DV80 files. It sucks a big chunk of the file into memory, displays the file, sucks more into memory, etc., indefinitely. Speed Reader's text buffer is larger than that of any 40-column version of TI writer. Bruce Harrison says he developed Speed Reader in part so he could display on screen the large (more than 200 sectors) Sherlock Holmes book files I have made available to the TI community.

Bruce has decided to get rich quick with this software. Unlike a lot of his offerings, it is commercial rather than public domain. I can't send it to you. To get your copy on a SSSD disk you need to send $5 directly to Bruce. This includes the cost of shipping and media.

P-GRAM RELOADER v1.0
by Tony Knerr

This will reload the entire contents of your P-GRAM+ card all in one smooth continuous operation. All four GRAM banks and the DSR are loaded in this process. If you normally keep the same software in your P-GRAM then you will like P-GRAM Reloader. You can experiment with your P-GRAM, temporarily loading in some different software. Later you can restore your original P-GRAM software suite easily with P-GRAM Reloader. This is particularly useful if you want to experiment with a module that uses module RAM at >6000. Doing this will erase the version of extended basic that almost everyone keeps in bank one of their P-GRAM+. You can then quickly use P-GRAM Reloader to restore your P-GRAM's normal extended basic. P-GRAM Reloader is also useful if you think your P-GRAM's memory has been corrupted.

P-GRAM Reloader will work from floppy, RAMdisk, or HFDC-controlled hard drive. You have to put all your P-GRAM files on the same floppy, RAMdisk or hard drive directory. Working from floppy probably requires a DSDD-formatted disk to hold all the P-GRAM files. This means you need a double density (not an official TI) disk controller. If you have a TI controller you can keep all your P-GRAM files on a RAMdisk.

Setting up your P-GRAM files for use by Reloader is a bit tricky, but you will only have to do this once. You copy the P-GRAM DSR and all your GRAM files to the working disk or directory. You then rename most of the GRAM files, following the instructions in Reloader's documentation. Finally, you have to use a sector editor to change the header of one of the GRAM files.

P-GRAM Reloader runs from Editor/Assembler Option 3 or TI BASIC. TI BASIC use requires a CorComp or Myarc floppy disk controller. Just run the software and your P-GRAM will be reloaded, overwriting everything already in the P-GRAM's memory.

This is software that every P-GRAM+ owner should have. It is not meant for the regular, non plus, P-GRAMs that only have one bank of GRAM memory and it is not for GRAM Krackers, Gramulators, or other GRAM devices. P-GRAM Reloader is public domain. Send me $1 and I will mail it to you on a SSSD disk.

ASSEMBLY POKER
by Marcel's Software

Marcel Barbeau is Bruce Harrison's son. Sales of this game augment Marcel's allowance. The software was actually written by Bruce Harrison. It boots from Extended BASIC or EA5.

This is one of the best poker games I have seen on the TI. There is some skill and mystery involved in this game for one player against the computer. You don't know what your computer opponent's hand looks like until the hand of poker is over, and if you fold you never find out.

The rules of five-card draw poker game play are literally "according to Hoyle." This is a one-player game. You play against the computer, whose cards you don't see until the end of the hand. Your cards are displayed at the bottom of the screen in sorted order with a printed statement telling you about your hand ("ace high," or "pair of 10s").

You and the computer each start the game with a $200 pool of money. Each hand starts with you and the computer putting a $4 ante into the pot. You each get the opportunity to bet and "see" one another's bets. At this stage, the computer player may bluff on a garbage hand, since this hand can be improved later by drawing replacement cards.

`When replacement cards are offered, if the computer player takes fewer than three cards this means that it has a good hand. If it takes no replacements, this means it has at least a straight. The number of replacement cards taken by the computer player is really the only clue the human player has concerning the contents of the computer's hand. After replacement cards an additional round of betting occurs. This time the computer player will not bet highly on a bad hand and may fold.

The human and computer player start a session with $200 each. The game ends when either player has less than the $4 ante for a new hand. However, you can end the game early by having the human player answer "N" to the "Another hand?" prompt. Thus, Assembly Poker makes an excellent short TI game. If you have a few minutes to kill just slip the SSSD disk into DSK1. The software boots quickly and you can then play a few quick hands of poker.

The game is fun with realistic play action, and the graphics are well done. There is no music or speech, but good poker players like it quiet so they can concentrate on their game. Assembly Poker is commercial. To get it send $5 to Marcel Barbeau. This includes the cost of shipping and handling.

AMS VIDEO TITLER
by Bruce Harrison

I have previously reviewed Video Titler, software designed to create neat videotape title sequences by artfully changing between in-memory full screen graphics. The original let you simultaneously load two full screen graphics into memory. This version lets you load in up to 41 screens, storing them in the AMS card's memory.

You can load TI-Artist "_P" pictures with or without their "_C" color file and/or you can load color pictures created with Harrison's Drawing Program. You

(See Page 37)

MICROREVIEWS —

(Continued from Page 36)

load these one at a time, specifying the path and file name for each when prompted. Without an AMS card you can fit only two of these graphics into memory, which can make using Titler quite cumbersome. With the common 256K AMS card you can load 20 graphics. The 128K AMS card will let you load 9 graphics and if you have upgraded to a 512K AMS card you can load up a whopping 41 frames. Once this is done you just press a key to zip from one graphic to the next in sequence. At any time you can pause the VCR and load more pictures, replacing your choice of those already in memory or filling as yet unused memory. Such changing pictures in memory in the middle of a recording session should not be necessary with the AMS version of Titler. It is almost always required if you use the non-AMS version limited to only two in memory graphics.

There are a zillion different neat ways of wiping the screen from one graphic to the next; slow or fast, top to bottom or bottom to top, corner to corner, edges to center or center to edges, etc. Each of these methods for paging through the graphics in memory is controlled by a single keypress.

If you don't want to make videotape titles you can just use AMS Titler to page through libraries of TI-Artist pictures. This would result in a really neat business or user group video presentation.

The software has an XB loader but is really an EA5 program that resides in low memory. It uses high memory to store graphics. The AMS card can be set up to bank 4K blocks of high memory. Each graphic uses three of these 4K memory banks. Video Titler displays in bit map mode, dumping the contents of the desired graphic into VDP RAM.

AMS card owners have been asking for more useful AMS-capable software. Here it is! Bruce Harrison tells me that anybody with enough money to buy an AMS card should be willing to pay a little for AMS capable software. AMS Video Titler is not public domain. You can buy your copy from Bruce on a DSSD disk for $10. This includes shipping, handling, seven samples of on disk graphics to load up, and over-the-phone product support.

CASINO (KLONDIKE) SOLITAIRE
by Ken Gilliland and Notung Software

This can be played from extended basic all by itself right off the disk, or it can be added as a module to Notung's TI Casino suite. The game, labeled Klondike Solitaire when played as part of TI Casino, differs little from your basic one-deck one-player generic solitaire.

Klondike Solitaire is almost identical to the solitaire that comes bundled with Windows 3.1 played on an IBM-compatible, except that in Klondike Solitaire there is some wagering. My wife sometimes spends hours playing the Windows version of solitaire on her IBM. It is one of the ways she relaxes after a hard day at work and dealing with the spouse and kids. Solitaire is a very enjoyable, slow, easy way to kill some time. There is some skill involved and a single game doesn't take very long. You can play just one quick game or spend several hours trying to build up your winnings.

You start off by buying a deck of cards. There are cheap, medium and expensive decks. The more you pay for your deck, the greater are your potential winnings. If you are playing the standalone game, without the TI Casino interface, you are given your choice of either $500, $1,000, or $5,000 at the beginning of the game. When the game ends because you run out of money or because you get tired of playing, then you go back to Extended BASIC command mode and all your money disappears. If you are using the TI Casino interface you can save your winnings in an account for use in a later game, or you can literally write yourself a check for any surplus funds you have accumulated.

The computer starts each game by shuffling the deck and dealing some cards out into card stacks. The remaining undealt cards make up the "deck" from which you may draw one card at a time as needed. The top card in each card stack is exposed. You can arrange cards in these stacks in decreasing numerical order using alternating colors. The object of the game is to build four suit stacks in numerical order starting with ace (on the bottom of each suit stack) and ending with king. These suit stacks are in separate piles, different piles than the card stacks. You may move any exposed card from the card stacks or the most recently exposed card from the deck to any of the four suit stacks if that card is the next in the numerical order suit pile. You are allowed to go through the "deck" a maximum of three times per game. The game ends when you have completed all four suit stacks or when you can't add any more cards to the suit stacks. Your winnings are computed based on the number of cards in your suit stacks, $x per card. The money awarded per card depends on how much you paid for your deck at the beginning of the game. You win some money in almost all of the games you play. The trick is to win back more than you initially paid for the deck. When a game ends you can keep your money and exit Solitaire or shuffle the deck again and play another game.

As in all the TI Casino games, a mouse is required. You move the cursor over a card you want to move and press the fire button. Use the joystick handle to drag the card where you want it and press the fire button to drop the card. The computer won't let you cheat. You can't drop a card on a pile where it isn't supposed to go. You also use the joystick to uncover cards in the "deck" by placing the cursor on the deck and pressing the fire button to display the next card. Joystick use in Klondike Solitaire is almost identical to mouse use in Windows 3.1 solitaire.

If Solitaire is your favorite game, then it is best played directly from the disk without incorporating it into TI Casino. The game boots much faster when played outside of the TI Casino environment.

Configuring Klondike Solitaire into TI Casino, a separate product, also by Ken Gilliland, is a bit tricky. You have to follow closely the instructions in the Solitaire documentation. If you already have the original TI Casino and the extra cost extra features TI Casino Supplement, then adding Klondike Solitaire will almost completely fill a DSSD disk. In fact, you have to remove all the joke files except one in order to fit everything onto a DSSD

(See Page 38)

MICROREVIEWS —

(Continued from Page 37)

disk. TI Casino plus the Supplement plus Klondike Solitaire is not hard drive friendly. It is difficult to configure all the files for long path names.

When you add Klondike Solitaire to TI Casino you completely reconfigure the upstairs of the casino, adding rooms for additional casino games. These additional, as yet unreleased, casino games include five-card stud, wheel of fortune (different from roulette), horse racing, single-deck blackjack and paigow poker. When you click on any of these game rooms you are told that the game isn't installed and are returned to the upstairs casino lobby. I look forward to these additional games and will probably buy them all, because the TI Casino suite is my personal favorite 99/4A and Geneve entertainment software. I suspect that adding any more games to the existing TI Casino suite will require either a DSDD compatible disk controller card, a large RAMdisk, or a convenient software method of running the entire package from two drives.

Klondike Solitaire is commercial and sells for $7 plus postage. It is available from Notung Software, Ramcharged Computers and other TI dealers. These dealers also sell other parts of the TI Casino series.

ACCESS

Harrison Software, and Marcel's Software (Speed Reading, Assembly Poker, AMS Video Titler): 5705 40th Place, Hyattsville, MD 20781. Phone (301) 277-3467

Notung Software (Klondike Solitaire and other games in the TI Casino suite): 7647 McGroarty St., Tujunga, CA 91042

Charles Good (source for P-GRAM Loader): P.O. Box 647, Venedocia, OH 45894. Phone (419) 667-3131. Preferred e-mail good.6@osu.edu (other previously published e-mail addresses are still valid until the end of 1996).

USER NOTES

User Note correction

Two errors were found in a User Note published in the September/October 1996 MICROpendium. They occurred in the item titled "The importance of pitch and slope with speech and TEII."

The first error occurred in the second column on page 37. The line should read:
20 PRINT #1:"// 50 160"

The line had been printed as 20 PRINT #1:".. 50 160"

The second error occurred in the second column on page 38. The line should read:
60 PRINT "INTERNAL CODE IS: ":B$

The line had been printed as 60 PRINT "INTERNAL CODE IS: ":BK$

More functionality from Telco

This item has appeared in several newsletters. We're not sure who wrote it.

Using FCTN period in Telco puts a dividing line at the bottom of the screen so you can type text and actually see what you have typed before sending it in conference.

Another use for FCTN period is to use it to log onto Compuserve, Delphi, and other services. You can see what you type before you send it.

Another use of FCTN period is on Compuserve. If you want to download a file in a certain library do a directory of the library to see the particular file you want to download. Here is where FCTN period comes in handy. Press FCTN period if you already haven't done so already. After you see the directory and see the file you want, press "B" to get back to the menu screen. There is a time lapse before it gets back to the menu. What I do while I am waiting is to type the filename. You will see that entered at the bottom of the screen. Now, when the menu gives you a prompt, just press Enter and it will send the filename you have typed. It saves a few seconds. If you use Compuserve and Delphi as much as I do, it may save you a bit of money, too.

Getting more from disk report

Printing out a "disk report" from Disk Utilities v4.1 complete with comments can take a lot of paper, even if you use compressed print. If the disk has many files, the disk report will probably not fit onto the front of the disk envelope. Using System Setup, and the following special characters, you can print your report in compressed, subscript, double-strike with small linefeeds. From the Printer Setup submenu, type "S" next to Disk Report and then enter the following special character codes: 1B0F1B53011B330B.

You can refer to the documentation that comes with Disk Utilities to make this a permanent change so that it will always be available. The code above works with Epson-compatible printers.

Formatter linefeed fix

The following was written by Col Christensen and appeared in TIBUG, the newsletter of the Brisbane User Group.

I read somewhere of a fix for the painful habit the TI-Writer formatter has of printing three blank lines before starting the actual file. I guess this is all right for those who follow the intended procedure and load their printer with one whole page wasted so that the first line of perforations is just above the print head. Being of a rather frugal nature, I don't like to waste a full page of paper each time I do a formatted printout and, besides, a lot of my printer output involves single sheets. So, in which newsletter did I see that vital "fix?" I wouldn' have a clue, it was so long ago.

Well, the next best thing to do was to find the crucial value in the formatter file that caused the three blank lines to be printed. Easier said than done. I changed in turn every >0003 to a >0000 in the formatter code, but none produced the required effect.

I don't know exactly why, but I then tried changing the values of >0002 to

(See Page 39)

USER NOTES

(Continued from Page 38)

>0000. That's where I struck gold, silver anyway. Changing a certain value in the code from 2 to 0, printed only one blank line to begin, while changing the value to 20 made the formatter print a great number (I didn't bother to count.)

It seems that in the formatter print line subroutine a linefeed is printed before a text line is printed. Remember, the output device name of the formatter defaults to PIO.LF with linefeeds suppressed. So we're stuck with one linefeed at the beginning, but that's an improvement over three.

The sector of the FORMA1 (or FO with Funnelweb) disk file to change is the 26th or >1Ath from the beginning of the file. If you have saved FO and FP to a newly formatted disk, the sector to edit is >3C. The word to change is >56 from the beginning of the sector. The printout of the appropriate Disk-Aid screen display is shown in Fig. 1. Use the CHange command by typing CH 3C from the command mode. Once the screen display begins, press FCTN 4 to break, then type 56 then 0000, then Y N and Y to the remaining prompts.

Fig. 1

```
Drv2 NEWDISK     Sze>2D) Sec>3C
Addr 0 1  2 3  4 5  6 7
     8 9  A B  C D  E F
 00  2002 0420 C696 0580   .. .,..
 08  06C1 0420 C696 0200  ... ....
 10  2009 CA80 0056 04C0   ....V..
 18  DA80 007C 0760 22CC  ...¶.°".
 20  1304 0420 C6AA 0008  ... ....
 28  13A7 0747 16AE 045B  ...G...[
 30  CE4B C820 2288 229A  .K. ".".
 38  0760 22CA 1606 0420  .°"....
 40  C404 229A 3FE2 22DE   ..".?.".
 48  22CC 05A0 2288 0420  "...".. 
 50  B87C 22E2 0203 0002  .¶".....
 58  06A0 BA62 0203 0005  ...b....
 60  C803 22A2 0460 B080  ..".."..
 68  CE4B C0C3 130C 0201  .K......
```

Cecure changes phone number

Cecure Electronics has moved to new facilities in Muskego, Wisc., and also has a new telephone number. The new number of the company's technical support is (414) 422-994A (414-422-9942). It's fax line is (414) 422-9889.

The company's toll free order line remains (800) 959-9640. Cecure also has a BBS at (414) 422-9669.

A small program with a big punch

The following small program and article appeared in TI*MES, the newsletter of the TI99/4A Users Group of the United Kingdom.

This program can cause severe migraine headaches and possibly epileptic seizures in anyone who has a tendency toward same, especially migraine which is optically induced, when viewed on a 14-inch screen from about two feet away. Do not use this pattern in game programs.

```
10 CALL HCHAR(32,"AAAAAAAAAA
AAAAAA")
20 CALL COLOR(1,2,16)
30 GOTO 130
```

Technically, the difficulty crops up when the frequency of the vertical lines is three cycles per degree of visual arc, and the contrast is about 70 percent.

This program has a health warning, please heed. In the event of any visual distress, avert gaze quickly and break the program. No responsibility is accepted for any damage or injury. The pattern is used as part of a diagnostic process. It is based on articles published in *Brain*, No. 107, p. 989-1017, "A neurological basis for visual discomfort," 1984; *Archives of Neurology*, October 1989, No. 46, p. 1129-1132, "Migraine and stripe-induced visual discomfort."

Now if this doesn't make you curious, nothing will.

Eicher creates Web page

Daniel Eicher has created a new World Wide Web page on Delphi at www.delphi.com/people/eicher.

Currently, the page has two files. The first is a compendium of programming information, related to TI assembly and the Geneve "for those folks reading M. Zapf's articles," he says. The second is a file explaining how to access memory on a variety of third-party memory devices for the TI.

"Both of these documents are ongoing work — and any additional information people wish to send for inclusion would be appreciated," Eicher adds.

Bush is TI server

Mattie Bush of the Dallas TI Users Group has announced that she is a TI server on Startext to the Internet. For information, contact mattie@startext.net.in.

CLASSIFIED

FOR SALE

FOR SALE

Working TI99 Geneve 9640 system with Color Monitor, Keyboard, Two Disk Drives, software, Handbooks, and Smith Corona TPII daisy wheel printer. Come and get it all for $100. Ben. 512-255-6875. v13n6

TI EQUIPMENT

2- P-Boxes full — a bargain at $150 each. Several consoles and misc software included. All equipment sold as is. $275 for the whole lot plus $10% shipping. Contact Lewis T. Turner, 212 West Cramer Ave, Walkersville, MD 21793 v13n6

Devoted to the TI99/4A since 1984

Subscription Fees

❑ 6 issues, USA, **$35**

❑ 6 issues, Mexico, **$40.25**

❑ 6 issues, Canada **$42.50**

❑ 6 issues, other countries surface mail, **$40.00**

❑ 6 issues, other countries, air mail, **$52.00**

Outside U.S., pay via postal or international money order or credit card; personal checks from non-U.S. banks will be returned.

Address Changes

Subscribers who move may have the delivery of their most recent issue(s) delayed unless MICROpendium is notified six weeks in advance of address changes. Please include your old address as it ap-

Check each item ordered (or list on separate page) and enter total amount here:_____

Check/MO ❑ ❑ ❑ (check one)

Card No. _____

Expiration Date _____
(Minimum credit card order is $9)

Signature _____
(Required on credit card orders.)

No sales tax on magazine subscriptions. Texas residents add 7.75% sales tax on other items, including back issues and disk subscriptions. Credit card orders add 5%.

Name _____

Address _____

City _____

State _____ **ZIP**

The set of numbers at the top of your mailing label indicates the cover date of the last issue of your subscription.

Disks, Etc.

❑ **Back Issues**,$3.50 each to March 1996, later $6 each. List issues: _____

No price breaks on sets of back issues. Free shipping USA. Add $1, single issues to Canada/Mexico. Other foreign shipping75 cents single issue surface, $2.80 airmail. Write for foreign shipping on multiple copies.

OUT OF STOCK: Vols. 1, No. 1-2; Vol. 2, No. 1

❑ **MICROpendium Index** (2 SSSD disks, 1984-1992), Extended BASIC required $6.00

❑ **MICROpendium Index II** (9 SSSD disks — 1 for each year — 1984-1992), XB required $30.00

❑ **MICROpendium Index II with MICROdex 99** (11 SSSD disks), XB required $35.00

❑ **MICROdex 99** (for use with MP Index II, 2 SSSD disks), XB required $10.00

❑ **MICROpendium Index II** annual disks ordered separately (1 disk per year, 1984-1992); each $6.00

MICROdex 99, by Bill Gaskill, is a collection of programs that allow users of MP Index II to modify their index entries, as well as add entries. MICROdex 99 supports many other functions, including file merging, deletion of purged records, record counting and file browsing.

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 2.21** (req. DSSD or larger (for floppy & hard drive systems) $4.00

❑ **GPL 1.5** $4.00

❑ **Myarc Disk Manager 1.50**.....$4.00

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

❑ **Menu 80** (specify floppy or hard disk versions(s); includes SETCOLR, SHOWCOLOR, FIND, XUTILS, REMIND.....$4.00

GENEVE PUBLIC DOMAIN DISKS

These disks consists of public domain programs available from bulletin boards. If ordering DSDD, specify whether Myarc or CorComp.

	SSSD	DSSD	DSDD
❑ **Series 1**	$9.00	$7.00	$5.00
❑ **Series 2**	$9.00	$7.00	$5.00
❑ **Series 3**	$9.00	$7.00	$5.00
❑ **Series 4**	$9.00	$7.00	$5.00
❑ **Series 5**	$9.00	$7.00	$5.00
❑ **Series 6**	$9.00	$7.00	$5.00

SECOND CLASS

A 1/97 T
CHARLES GOOD
P.O. BOX 647
VENEDOCIA OH 45894