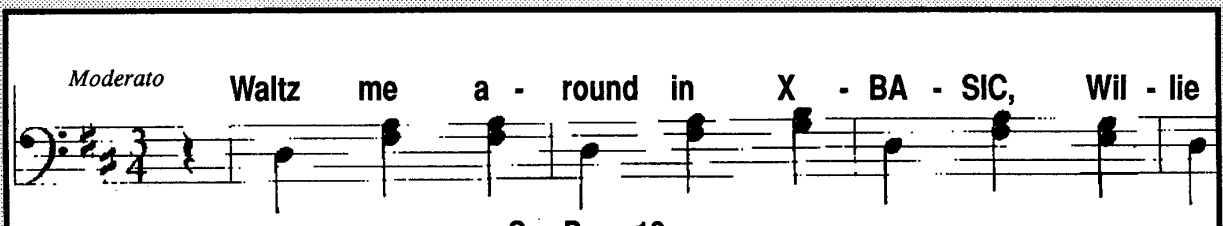
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 13 Number 3 May/June 1996 $6



See Page 18

Working with printers

Invisible characters in filenames

Limits of a random number algorithm

Reviews of:

√ Chainlink Solitaire *√ Virus Attack*

√ Word Search

√ Printer's Apprentice and Tool Box Tutorial

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. Second-class postage paid at Round Rock, Texas. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

No information published in the pages of MICROpendium may be used without permission of the publisher, Burns-Koloen Communications Inc. Only computer user groups that have exchange agreements with MICROpendium may excerpt articles appearing in MICROpendium without prior approval.

While all efforts are directed at providing factual and true information in published articles, the publisher cannot accept responsibility for errors that appear in advertising or text appearing in MICROpendium. The inclusion of brand names in text does not constitute an endorsement of any product by the publisher. Statements published by MICROpendium which reflect erroneously on individuals, products or companies will be corrected upon contacting the publisher.

Unless the author specifies, letters will be treated as unconditionally assigned for publication, copyright purposes and use in any other publication or brochure and are subject to MICROpendium's unrestricted right to edit and comment.

Display advertising deadlines and rates are available upon 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
GEnie: J.Koloen
Internet E-mail: jkoloen@io.com
John Koloen.....Publisher
Laura Burns.....Editor

The Art of Assembly

The limits of a random number algorithm.....Page 9

Printers

Using the Hewlett-Packard printer command language.....Page 13
Downloading user-defined fonts.....Page 24

Hidden Characters

Using invisible characters in filenames.....Page 16

Extended BASIC

Waltzing your way to user-defined subprograms.....Page 18
Disk catalog labeling.....Page 28

Hardware

'Stealing' power for external disk drives.....Page 21

PC99

Stage 3A upgrade includes host of new utilities.....Page 22

Reviews

Chainlink Solitaire.....Page 32
MICRO-Reviews: Virus Attack, Word Search, Printer's Apprentice and Toolbox Tutorial.....Page33

User Notes

The TI99/4A typewriter, adding characters in condensed mode, and using CHARA1FIX with Funnelweb characters.....Page 36

Classified.....Page 39

*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.

COMPETITION COMPUTER
350 MARCELLA WAY MILLBRAE, CALIFORNIA 94030
800-471-1600
TI99/4A CD ROM
(Nationwide & Canada ORDERS ONLY)
STILL NO EXTRA CHARGE FOR CREDIT CARD ORDERS
NO C.O.D.'S
NON-PROFIT SHIPPING CHARGES YOU ONLY PAY WHAT WE PAY
WE ACCEPT PERSONAL CHECKS & BUSINESS DAYS FOR CLEARANCE
MONEY ORDERS
MASTERCARD AND VISA
CD ROM UPDATE: CAN NOW BE USED WITH PC99 STAGE 3A USING NEW PC99 UTILITY
DLCONV.EXE / 5 ISSUE SUBSCRIPTIONS $10.00
SINGLE (120 MB) ISSUES NOW AVAILABLE $35.00

Picasso

One of the best!

Picasso 2.0

Picasso is an excellent drawing program that lets you create huge screens up to 480 x 342 in size - over twice the size of *TI Artist*! The latest version features font capabilities, icons, new window functions, *TI Artist* file importing, and many other new functions, along with the excellent zoom and array of drawing functions found in the original. Great for creating graphics for *Page Pro 99* - you can draw up to 1/2 of a page at a time! Includes as a bonus the book "Using Picasso" - a $5.95 value, and a *Page Pro* converter. Requires 32K, Disk, Extended BASIC or Editor/Assembler and an Epson compatible printer.

G9101 Disk $14.95

For a limited time only! Picasso 2.0 *and all its companions together for just $29.95! Order Item* G9105

Picasso Borders

Borders for the *Picasso* drawing program. *Picasso Borders* contains over 50 great borders! By Paul Schleldenmantle. Requires *Picasso 2.0*, Disk, 32K.

G9102 Disk $5.95

Picasso Enlarger

A program to enlarge and reduce *Picasso* pictures, "ghost" a picture (make it appear glitterly), and import *TI Artist* instances directly into *Picasso* pictures. This utility package is a "must" for *Picasso* users! Requires *Picasso*, 32K Extended BASIC and disk.

G9103 Disk $7.95

Picasso Utilities

This package of utilities will let you get more out of *Picasso*! *Picasso Utilities* includes an Extended Font utility that prints "camera ready", high quality copies of your *Picasso* screens, as well as a utility for typing on your *Picasso* screens in large fonts. Requires *Picasso*, 32K, Extended BASIC, Disk and an Epson compatible printer.

G9104 Disk $7.95

TI-Artist & Artist Plus

When you are on the edge...

Artist Borders

For the first time, easily add borders to your *TI Artist* or *TI Artist Plus* screens! Each volume in the 5 volume *Artist Borders* collection contains between 30 and 40 finely detailed borders for use in certificates, signs, etc. Requires *TI Artist* or *TI Artist Plus*, 32K and disk are required. A printer is recommended.

G9106a-e Disk *each - all 5 for $19.95 - Order* G9109

Artist Fonts

Now you can purchase fonts by themselves in our outstanding *Artist Fonts* series! These fonts were all drawn by hand by some of the best 99/4A artists today! All require a disk system & *TI Artist*.

G9110a-e Disk *$4.95 each all 5 for $19.95 Order* G9110

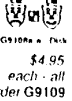
Sports Pics

An great collection of sports oriented pictures by Paul Scheldenmantle for use with *TI Artist* or *TI Artist Plus*. Requires 32K and a disk system.

G9114 - Disk $5.95

Artist Instances

This outstanding series of works is the most extensive library of clipart ever produced for *TI Artist*! Each volume consists of large, detailed and well-documented pictures stored in *TI Artist* "instance" format - ready for use. Arranged by theme - only purchase the packages that you want. Requires disk, 32K and *TI Artist* compatible software.



G9111a Volume 1 Animals 1 $4.95
G9111b Volume 2 Animals 2 $4.95
G9111c Volume 3 Holidays 1 $4.95
G9111d Volume 4 Miscellaneous $4.95
G9111e Volume 5 People 1 $4.95
G9111f Volume 6 Computers $4.95
G9111g Volume 7 People 2 $4.95
G9111h Volume 8 Animals 3 $4.95
G9111i Volume 9 Holidays 2 $4.95
G9111j Volume 10 Miscellaneous $4.95
G9111k Volume 11 Holidays 3 $4.95
G9111l Volume 12 Holidays 4 $4.95

Any 3 volumes only $11.95 - all 12 only $44.95! 10% off! *Order Item* G9112

Artist Enlarger

Artist Enlarger will enlarge or reduce, stretch or squeeze any *TI Artist* compatible font or instance, of any size. Menu driven and simple to use - this utility is essential for the serious graphics user! By Howard Hoban. Requires 32K, Disk and Extended BASIC.

G9113 Disk $5.95

Make or modify TI-Artist fonts!

Artist Font Maker

This utility allows you to literally convert any picture into a character of a *TI Artist* or *TI Artist Plus* font. Now for the first time you can create a font from scratch within *TI Artist* or in any other drawing program, and then load it into this program, and "dump" each letter to a font file of your creation. The beginning graphics user will appreciate the ease of use and the powerful features that make creating a new font simple. All users will appreciate the tremendous latitude the program affords in font generation. Requires 32K, disk and either the Editor/Assembler or Extended BASIC modules. By Jim Reiss and Asgard Software. All As in *MICROpendium!*

G9115 Disk $9.95

Make fonts, banners, and more!

Font Writer II

The ultimate companion for *TI Artist* users - this package contains a *font editor*, a *font manager* for editing and merging fonts, a *banner maker* for creating banners with CSGD fonts and Artist instances, *conversion utilities* for converting between *TI Artist* and CSGD, and a TI Writer like *font formatter* that allows you to freely mix printer and *TI Artist* font text with pictures and borders! Even do simple desktop publishing! Requires 32K, Extended BASIC and a disk system. By Peter Hoddie

G9116 Disk $12.95

GRAPHX

A classic!

GRAPHX Companion Series

This library of 4 different comp-anions for use with *GRAPHX* contains 40 fonts and hundreds of pieces of clipart, animation sequences, borders, fill patterns and more for use with *GRAPHX*. 7 disks of art at one low price! Requires *GRAPHX*, 32K and a disk system.

G9119 Disk $16.95

GRAPHX Slideshow

An easy to use slideshow program - catalogs a disk and displays all of the pictures in sequence at the rate you specify. Requires 32K, a disk system, and *GRAPHX* format pictures. By Paul Charlton.

G9120 Disk $5.95

Converters

Pix Pro

The "Super Converter" to meet all your graphics needs! There are many graphics formats in use today - many times pictures are available in one format but can't be found in another. Now, for the first time, this simple to use utility gives all your programs access to the entire library of pictures available for the 4A and 9640! With *Pix Pro* you can convert pictures stored in *GRAPHX*, *TI Artist Picture*, *Instance*, *RLE*, *Picasso*, *Page Pro*, *MacPaint* and other formats into most any of the above. Store a *MacPaint* or *Picasso* picture in *TI Artist* Picture format, convert a *GRAPHX* screen to Instance format, go from *RLE* to *Picasso*, and most any other conversion you can dream of, all easily and at Assembly speed! *Pix Pro* will even let you save a full page *MacPaint* as a full page *Page Pro*! Full color support is included. The essential tool for all desktop publishers! Requires 32K, Disk, Extended BASIC or Editor/Assembler. By Jim Reiss.

G9121 Disk $14.95

Asgard Mouse

The only mouse available that works on virtually all 99/4A compatible systems, the *Asgard Mouse* is a high quality pointing device for the TI 99/4A and the Myarc Geneve 9640. Installed on any RS232 port in seconds - to use it simply load a compatible program and go! The *Asgard Mouse* is compatible with *TI Artist Plus*, *YAPP*, *Page Pro Poster Maker*, *Batch It!*, *Classic Checkers*, *TI Pet* and other programs. A *TI Artist/Artist Plus* driver, along with demonstration programs and an extensive manual is included. Requires an RS232 card, disk system expanded memory.

G9121 Disk $39.95

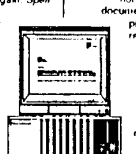
Word Processing

The best for the 99/4A!

Spell It!

New Version!

With *Spell It!* you'll never have to worry about embarrassing spelling mistakes in your TI Writer or compatible documents again. *Spell It!* makes checking your spelling on the 4A easier than writing the words in the first place! Why is *Spell It!* the best? *Spell It!* is many times faster than any other TI spelling checker. It lets you view the context a misspelled word is used in. It allows you to add words to your user dictionary (limited only by available disk space) as you check your text files. It has no limit on text file size. It lets you look up a word in the dictionary if you aren't sure of a spelling. It creates a copy of your text file with all the corrections already made, and it even supports foreign languages. *Spell It!* is available in three versions - SS/SD and DS/DD disk versions with over 25,000 words, and a hard disk version with over 250,000 words. *Spell It!* requires 32K, one SS/SD disk drive and TI Writer or a compatible. Provided unprotected and RAM disk compatible.



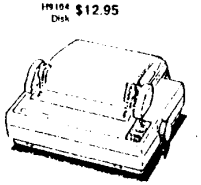
H9101 DS/DD Disk $14.95
H9102 SS/SD Disk $19.95
H9103 HFDC Version $29.95

A paper saver!

Screen Preview

Do you want to know how to use the TI Writer Formatter? Are you frustrated at not being able to see how your document will come out before it is printed? *Screen Preview* is a replacement for the TI Writer (or compatible) Formatter. As the name suggests, this program formats your document to the screen as dots on a representation of the page. With this you can easily check centering, margins, justification and so forth before printing. You can even make changes directly in your formatted text! Simple to use and extensively documented, this program will save you mounds of paper and hours of frustration. It is a great educational tool as well! Requires a TI 99/4A or Myarc Geneve 9640, 32K, a disk system, a printer, and TI Writer or a compatible. Written by Joe Delekto.

H9104 Disk $12.95



Quick notes and envelopes!

Typewriter 99

Typewriter 99 is the remarkable program that turns your computer into a full featured electronic typewriter! A word processor is too much for some jobs - short notes, labels, addressing envelopes, etc. - that is why we wrote *Typewriter 99*. It features bold and underline commands, text justification and auto centering. Set margins/tabs on the screen, and even see the 8 lines previously typed. It's silly to buy a typewriter if you have a computer - with *Typewriter* you have everything you need! Both versions require a printer - the disk requires 32K. A 9640 version is included with disk version. By Jim Reiss.

H9105 Disk $14.95
H9106 Cartridge $19.95

Edit a disk of text at once!

LinEditor

Do you ever need to look at *REALLY* large text files, download large ASCII files from BBSs, write really large programs, or are you working on the Great American Novel? Ever run out of memory in TI Writer? If you answer to any of these questions is "yes", then you need a program that will let you load and edit a text file larger than what you can fit into the computer's memory. You need *LinEditor* - the fast text editor for the TI 99/4A or the Myarc Geneve 9640 that breaks the "what you can fit into memory" limit. This simple to use program makes viewing and changing big files a breeze. Simply load a file, tell it the filename you want to give the new version (you never lose your original - perfect for programmers!), and start typing, editing and deleting as you normally would. *LinEditor* requires a TI 99/4A with expanded memory and a disk system. It is compatible with the HFDC and RAM disks. By Edwin Hall.

H9107 Disk $14.95

School

For children ages 4 and up

Edu-Pack

A new educational cartridge by David Bishop, this module includes 3 educational games in one for children of all ages. *Letterland* is a game for young children that teaches the alphabet and numbers - the object is match the letter or number on your rocket as quickly as possible with the letters and numbers floating by. *Speedkey* is a typing speed game where you have to type the letter or number as it appears and before it travels across the screen. And finally, *Gixing* is a unique 2 player game that teaches cooperation by having players work together to plant a seed and grow food for the community. All three games are non violent, are colorful and easy to understand. A color display and a joystick are recommended.

H9108 Cartridge $ CALL

Business

Picture Calendars!

Calendar Maker 99

For the first time ever - create big, beautiful picture calendars on your TI-99/4A! *Calendar Maker 99* is a powerful program that MICROpendium magazine called "one of the easiest to use programs ever". Create a month or year calendar for any year from 1600 to 2400 A.D. with pictures and text on individual days, a big picture on the top of the page and a message on the bottom! Customize your calendars with different fonts and linestyles. Requires Extended BASIC, 32K, Disk and an Epson or compatible printer. By Chris Bobbitt and Ed Johnson.

H9109 Disk $14.95 *25% off!*

Enhance your calendars

Calendar Maker Utilities

A utility package that lets you do even more with *Calendar Maker 99!* Design new fonts and linestyles with this easy to use editor. Create header files from any *TI Artist* font in several languages. Create "Universal Data Files" with oft-used text so that you only have to enter in birthdays and holidays once. Even print out samples of all your fonts! Requires *Calendar Maker 99*, Disk, 32K and Extended BASIC. By Chris Bobbitt and Ed Johnson.

H9110 Disk $9.95

Manage your life!

Mail Room

The Personal Information Manager for the 99/4A and Geneve! *Mail Room* is designed around an easy to use mailing list database - simply enter in names, address and phone information about clients, friends or family, and you have the beginnings of your personal database. Want to call someone? No problem, turn on your modem and tell the computer to dial away! Need to get a subset of the list for some specific purpose (a mailing, etc)? Again, no problem. Want to generate form letters? Just output the records you want to disk in TI Writer "Merge" format and print. Need labels or envelopes in a jiffy? It's just a few keys with *Mail Room*. No programming required, *Mail Room* is a complete, self-contained program that gets you up and running quickly. *Mail Room* includes both 40- and 80-column versions for the 99/4A and the 9640. A disk system, expanded memory, and Extended BASIC are required. A printer and modem are optional, but recommended. By Larry Tippett.

H9111 Disk $14.95

Computer Rolodex

Total Filer

Total Filer is a free form database for text. Perfect if you want to index a lot of data about a collection of books, articles, or even a genealogy. *Total Filer* allows you to do a rapid keyword search of any item in your database. Great for research projects, or even as an electronic rolodex. By Warren Agee. Requires Disk, 32K and Editor/Assembler. Printer optional.

H9112 Disk $7.95

Never miss a date!

Schedule Manager

A sophisticated, easy to use appointment book for the 99/4A and Geneve! Features a full screen editor, Mac like interface, calendar and notebook for storing names and phone numbers! This powerful reminder function makes this an all around program for managing any schedule - or even multiple schedules! Well documented and easy to use. Requires 32K, Disk and Extended BASIC. A printer is recommended.

By Chris Bobbitt.

H9113 Disk $8.95

Cassette Labeler

A simple but useful utility for managing all your cassettes! This program lets you create cassette box inserts detailing the contents of your cassettes - place inside your cassette boxes for neat, attractive organization. Requires Extended BASIC and an Epson compatible printer. By Tom Wynne.

H9114 Disk $5.95

COMMENTS

We're bimonthly now

As you can see, we're now a bimonthly publication. Laura and Iappreciate all the encouragement we've received from you.

Just in case you didn't receive the postcard we sent, we will be publishing according to the following schedule:
May-June
July-August
September-October
November-December
January-February
March-April
et cetera.

Disk subscribers will be charged $25 for six issues of the MICROpendium disk. Those who have already paid $40 will have their disk subscriptions extended. You'll notice the new expiration date on the mailing label with the April-May disk. The cost of individual copies of the disk will go from the current $4 to $5. Individual hard-copy issues of MICROpendium go to $6.

TRAGEDY OF RED BARON

Once again tragedy struck the TI world. The death of Shawn Baron leaves a void in the TI community. Young Shawn, only 17 years old, died March 25 as a result of influenza.

He and his father, Richard Baron, had only recently started Red Baron Software when Shawn became ill. His death a week later ended the venture and denied us the benefits of what the young man and his father had dreamed of doing. We add our condolences to those of the TI community to the Baron family, as well as to the families of Frank Bunzel and Meredith Beyers, two long-term TIers who made a number of contributions to the community.

SURVIVAL GUIDE

The more we publish MICROpendium, the more I see it as a survival guide for TIers. That's why you're seeing articles and reviews that were published elsewhere years ago. It's so you'll have a reference if you ever need to find a piece of software that hasn't been published since 1987. There was a lot of stuff we didn't cover back then. Now we're catching up.

—JK

READER TO READER

Edward Babowicz, 410 Dawes Ave., Utica, NY 13502-1620, writes

I have a Gorilla Banana printer with an Axion interface. How do I get the double width character mode? An example would be fine. The second question is, do you know of any supplier that handles replacement ribbons for the Gorilla Banana?

Stanley Ulanoff, 113 N. Carondolet Court, Mobile, AL 36608, writes:

I am trying to hook up a 19-inch TV as a monitor to my TI/Geneve. I have tried using the old TI modulator but it was too grainy and snowy. I would appreciate any help I can get.

Jacques Groslouis, 1747 Riverbank Dr., Bathurst, New Brunswick, Canada E2A 4L1, writes:

I have put together a Companion disk for users of Funnelweb Editors v5.01 which allows the user to include instances and Business Graph 99 printouts in a Funnelweb document. In addition, the disk allows the downloading of various fonts directly when printing the document. Different types of CHAR@1 fonts and Peterson type fonts can be created and used in this fashion. My programs work well with my Epson FX-80 printer and I would like to identify and provide ways so that users of other Epson-compatible printers can use these programs with their printers. The main printer functions used in my programs are as follows:

<ESC>"%"CHR$(n1)CHR$(n2) which selects a character set; n1 selects ROM(0) or RAM(1); n2 is 0.

<ESC>"%"CHR$(n1)CHR$(n2)CHR$(n3) which copies ROM characters to the user RAM area. All numbers must be 0.

<ESC>"%"CHR$(n1)CHR$(c1)CHR$(c2)CHR$(A)CHR$(d1)...CHR$(d11)... which defines characters c1 to c2 in RAM area; n is 0. Each character requires an attribute byte(A), followed by 11 data numbers (d1 to d11).

I would appreciate receiving from owners of Epson-compatible nine-pin printers photocopies of their printer manual pages which describe the operation of the above commands on their printers. Please remember to identify your printer. Once I have identified some of the different commands, I intend to release my Companion disk as a public domain disk.

Reader to Reader is a column to put TI and Geneve users in touch with other users. Address questions to *Reader to Reader, c/o MICROpendium, P.O. Box 1343, Round Rock, TX 78680*. We encourage those who answer the questions to forward us a copy of the reply to share with readers.

TEX COMP LTD.

DON'T FORGET THAT WE STILL CARRY ONE OF THE LARGEST SHAREWARE LIBRARIES AVAILABLE FOR THE TEXAS INSTRUMENTS HOME COMPUTER WITH OVER 300 DISK TO CHOSE FROM. CHECK OUT THE SUPER DISK PRICES BELOW AND MAKE YOUR ORDER TODAY.

CAN'T FIND DISKETTES FOR YOUR TI ?

By special arrangement with the manufacturer TEX-COMP Ltd will be able to meet and fullfill your diskette needs well into the 21 st century.

**BULK DISKETTES*

SUPER BUY!

BLACK	.19 EA (MIN 50)
COLOR (R,O,Y,G,B,W)	.34 EA (MIN 50 PER COLOR)
"FLIPPY"	.45 EA (MIN 50)
"FLIPPY COLOR (R,O,Y,G,B,W,)	..50 EA (MIN 50 PER COLOR)

PAPER SLEEVES	ADD .02 PER DISK
TYVEC SLEEVES	ADD .04 PER DISK
LABELS	100 PER KIT $2.00

R=RED,O=ORANGE,Y=YELLOW,G=GREEN,B=BLUE,W=WHITE

CALL FOR QUANTITY DISCOUNTS

BOXED DISKETTES	*PER BOX*
**NAME BRAND	4.95
GENERIC***	3.75
COLOR***	4.25
RAINBOW***	5.50

BOX COMES WITH LABLES AND SLEEVES
*****ADD $1.75 PER BOX FOR FLIPPY DISKETTES**

BOXED DISKS

ADD .01 ADDITIONAL SHIPPING PER DISK

* PRICE AND AVAILABILITY SUBJECT TO CHANGE WITHOUT NOTICE!
** BRAND WILL VARY (WILL BE PREMIUM QUALITY DISKETTES)

4/A FLYER	$24.95	MIDNIGHT MASON	$24.95
BARRAGE	$29.95	MINI WRITER	$24.95
BEYOND PARSEC	$24.95	MUNCHMAN II	$24.95
BEYOND WORD WRITER	$39.95	PARAPRINT	$69.95
BLACK HOLE	$24.95	TI PLANNER	$24.95
BOXER	$24.95	PROTYPER	$24.95
BREAKTHGH/ BREAKOUT	$24.95	QMAZE	$29.95
BURGER BUILDER	$24.95	RED BARON	$39.95
COMPUTER WAR	$29.95	SORGON II	$29.95
CONCOLE CALC	$24.95	SPACE PATROL	$24.95
D-STATION I	$24.95	SPOT SHOT	$29.95
D-STATION II	$24.95	SPY'S DEMISE	$24.95
DRAGON FLYER	$29.95	STARGAZER	$39.95
DESKTOP PUBLISHER	$39.95	STAR RUNNER	$29.95
ESCAPE	$24.95	STAR TRAP	$24.95
FACE CHASE	$24.95	STRIKE 3	$29.95
GREAT WORD RACE	$29.95	TI TOAD	$24.95
JUMPY	$29.95	TI WORKSHOP	$49.95
JUNK MAN JR	$29.95	WORD WRITER	$29.95
MAGIC MEMORY	$49.95	WORD WRITER EXTRA	$29.95
MANCALA	$29.95		
MICRO PINBALL	$29.95		
MICRO TENNIS	$29.95		

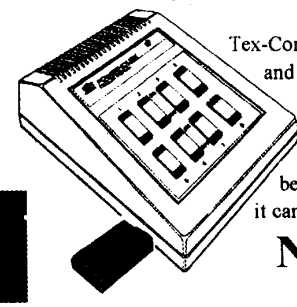
DBT DataBioTics

SOLID STATE CARTRIDGES

X-10 POWERHOUSE

Computer Interface for the TI-99/4A

Now you and your TI can control the whole house right at you desk. Using the controller to turn on the lights, water the lawn, turn on the TV and much more all at your finger tips. The powerhouse has a real time clock, battery backup in case of loss of power, & has 8 control buttons to allow you to manually control appliances and lamps. **No external storage or expansion memory is Required to use the X-10 Powerhouse!!**



Tex-Comp sold the X-10 system for $79.95 and we are bringing them back at the same price for a limited time. Since we last carried the X-10 system it has grown and is now estimated to be in 3 million American homes, now it can be in your home too.

NOW $79.95

SUPER AMS CARD

TAKE YOUR TI TO THE NEXT LEVEL!

This memory expansion card breaks the 32K barrier allowing for very large E/A programs or multiple XB programs using the included XB packer. The number of XB programs loadable at one time depends on the memory option ordered. Comes with modified assembler allowing use of all the new features and a multi-page manual on disk.

There are two versions to chose from

128K version	**$85**
256K version (recommended)	**$100**
128K to 256K upgrade	**$25**

ADD $5 EXTRA TO SHIPPING FOR INSURANCE, PER AMS CARD.

If you don't see it here call & we will attempt to source anything for your TI.

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 TECH. # (818) 339-8924 ADD 5% SHIPPING {$5.00 MINIMUM}

ALL PRICES ARE FOR CASH/CHECK, ADD 3.5% FOR CREDIT CARD ORDERS

FEEDBACK

About publication, bar codes for mail

First things first: Concerning your comments "Considering options" in the February 1996 issue, I talked to a manager of a large printing firm in northeast Ohio. He explained that for a limited run publication of several thousand copies, most large printers use a Xerox Dupli-tech machine. The content is entered from a disk. The machine formats, prints, collates and binds the material in one operation. There is virtually no manual intervention. The cost therefore is less than one half that of conventional printing. Not knowing how MICROpendium is printed, you may already be doing this. He also stated that although costs are going up, and increases in printing prices can be expected, there is no industry-wide reason why the cost should even come close to doubling.

Other alternatives: Possibly a local newspaper could do the printing less costly, much as they do an advertising supplement. Maybe you could do a survey of subscribers, as to age, income, other interests, etc. This information then could be used to attract new advertisers for a select group of persons in areas other than the 99 and 9640. Would contributors to the User Notes be willing to forgo the $10 payment?

I am sure many other readers also have a lot of ideas. With everyone pulling together, there will be a way for MICROpendium to continue as is.

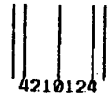
Now to comment on your review of TI-BAR in the January 1996 issue and the letter from Dr. Charles Good in the February issue. What was stated in both instances is absolutely true. However, mail will be processed as having a bar code if, and only if, the machines know that there is a bar code on the mail piece. A little background: The first step in processing first class mail is to face (orient the mail so all stamps are down, cancel (apply the postmark) and primary sort. The primary sort consists of three categories: A. Bar coded mail. B. Handwritten mail. C. Machine-readable mail (typed or printed). Separation into categories B and C is accomplished by cameras that look at the mail piece for certain format qualities. These qualities include left characters and equal spacing between lines. There is no provision to read a bar code in the address block on the facer-canceler machines. The only way that the facer-cancelers know that a bar code exists, and will sort to category A, is if there is an FIM code on the mail piece. The FIM (facing identification mark) code is the long bars on top of the mail piece, to the left of the indicia (stamp or meter mark). You have probably s[illegible] this code many times on mail, such as [illegible]turn envelopes for utility bills. Because the mail travels through these machines in excess of 12 feet per second, the size, placement and spacing of this code is extremely important. The DMM, available at any post office, gives the specifications.

So how much time do we save by barcoding our mail and sorting to category A with the proper FIM code? It all depends. Category A mail goes directly to the bar code sorters, bypassing the OCRs (optical character reader machines). This can save from several hours to one day, if the cutoff time for a dispatch is missed. As Charles Good pointed out, the OCRs may spray an additional bar code on the letter. This could result in extra delays, particularly if one bar code gets sprayed on top of another. With the advent of "wide area bar code readers," the preferred location for a bar code is above the name and address block.

This is by no means meant to criticize the TI-BAR program, only to offer a little insight on the workings of the postal system, as the author may wish to make a f[illegible] changes so we may all benefit more from its use

Lew King
Industry, Pennsylvania

For the technically curious.



The right-most bar must be from 1 7/8 to 2 1/8" from right edge of envelope.

Height should be from 5/8 to 3/4" with spacing of 1/16".

The FIM bars are a binary code as shown. Inactive bars are missing. The two outside bars are framing bars and not counted.

The bar code itself is a strange binary code. With a 7 in place of the normal 8. Any digits adding up to over 9 are counted as zero. As in the FIM code, the outside lines are framing bars.

None of the sizes or spacings are correct in this example. About the best I could do in the FW editor.

Marketer omitted

I have been following the series of Bill Gaskill's articles and am impressed with the chronology of events which took place in 1982 and 1983. But I would like to point out that he has omitted an interesting thing that also happened during that time frame.

In March 1982, I was introduced to the TI99/4A computer by a Tronics distributor. Tronics was a multilevel marketing company patterned something like Amway. Not only did my wife and I purchase a TI99/4A home computer, we also became a part of the multilevel marketing company. I really don't know how extensive the influence of Tronics became as I was going to college at the time and didn't have a lot of time for participation. But without Tronics, there would have bee[illegible] fewer TI computers sold. I doubt I would have given TI much time as I preferred to

(See Page 8)

MYARC

ANNOUNCING MYARC'S 640K RAM

GENEVE t.m.

MODEL 9640 FAMILY COMPUTER

This unit is without a doubt the most sophisticated machine ever offered in the family and small business area to date. With over a year of design and development, including input from more than one hundred users, this machine has surpassed even our own expectations. Take a moment to review some of the many features that place this computer in a class of its own.

- 99/4(A) COMPATIBLE RUNS OVER 100 EXISTING TI CARTRIDGE PROGRAMS
- TI-WRITER NOW A FULL 80 COLUMNS

LARGER
Standard 512K 1 Wait State CPU Memory
32K High Speed 0 Wait State CPU Memory
Expandable to 384K 0 Wait State
128K VDP Memory-Expandable to 192K
Memory Expansion Cards Compatibility Include:
MYARC 512K
RAVE 99 MX-01
MEMEX 504
2 MEGABYTES addressable memory
FASTER
At least 2-3 times than a standard TI-99/4A
MOUSE SUPPORT

- 99/4(A) COMPATIBLE RUNS OVER 95% OF ALL ASSEMBLY LANGUAGE PROGRAMS & UTILITIES
- MULTIPLAN ALSO 80 COLUMNS

AVAILABLE NOW

$199.95

- Separated Function Keys
- Incredible Graphics Capabilities
- Basic 3.0
- Composite Video Output
- RGB Output
- 128K VDP RAM Memory

- 40 Column Display
- 80 Column Display
- Mouse Output Port
- Joystick Port
- Sound Compatible

- Compatible with Existing Peripheral Cards* Disk Controllers* (MYARC, TI, Corcomp) RS232 Cards* (MYARC, TI, Corcomp)

- True Hardware Utilities Sprites, Fills, Lines, Data Moves
- TI 9995 Processor Chip - 12 MHz

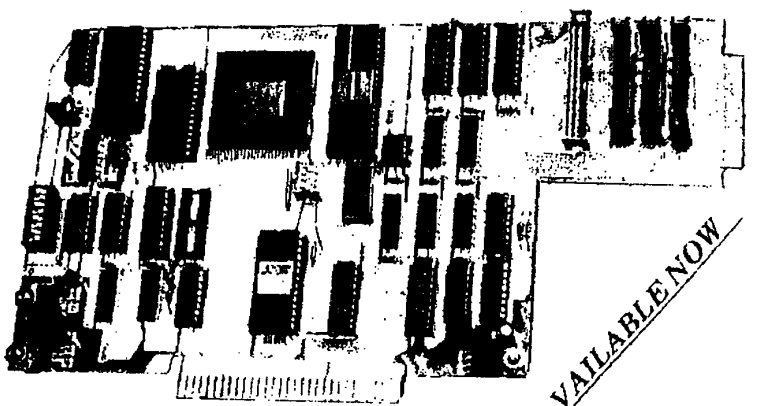
GENEVE 9640 TM

CECURE ELECTRONICS INC.

We accept Master Card and VISA

MasterCard VISA

THE MYARC Hard & Floppy Disk Controller



AVAILABLE NOW

$199.95

Features Include:

Hard drive transfer rate of 5Mbit per second, for speed comparable to an external RAM disk card.

Interfaces with standard, off the shelf, hard, floppy and streamer tape drives.

Built-in real time clock, for time and date stamping of files.

Supports up to four 5 1/4" and/or 3 1/2" floppy drives, mix or match.

All disk formats, SS/SD (90K), DS/DD (320/360K), DS/QD (640/720K) supported.

DS/HD (1.44MEG) supported on GENEVE 9640 ONLY.

MYARC Disk Manager V, the most intuitive and user friendly manager available.

WE'RE BACK!!!

MYARC TM

DIVISION of CEI

Make Checks Payable to:
CECURE ELECTRONICS INC.
P.O. Box 132
Muskego, WI 53150

ORDERS ONLY 800-959-9640

Technical Help 414-679-9918

FEEDBACK

(Continued from Page 6)

use Radio Shack equipment. I especially liked the Radio Shack Color Computer, and had a COCO 2. Through Tronics, I was exposed to the TI machine, bought one and was involved for a while with Tronics in the Fort Worth, Texas, area, then dropped out in late 1993. It was about a year ago that I spotted a TI99/4A in a Goodwill Thrift Store for $14, and bought it. One thing leads to another, and now I have the expansion box, 32K memory, disk drives and RS232.

Edward T. Verell
Williamsburg, Virginia

Exploring roots

I was reading with interest Lucie Dorais' Extended BASIC article on Roots in the February edition. The computerized "trial and error" method sounded interesting, so I was going to key in the program and then compare that speed to using the exponentiation function already built into BASIC/Extended BASIC. As I was typing, I discovered that's exactly what was being used, and the explanation was for a different method! This seems to be a much more direct approach anyway. With some of the validations removed, even fractional or non-integer roots may be taken.

I enjoy getting your magazine, regardless of size or frequency — just keep it coming regularly and keep it interesting!

Dave Erickson
San Jose, California

Hang in there

Reference your February 1996 editorial about "Considering Options," I would gladly pay $50 per year for the privilege of having a national magazine devoted to my favorite hobby computer. However, since I would guess that I'm in the minority on that option, I think six issues per year is the best long-term solution. Your idea about nine issues per year has merit, but I fear it is only a temporary move that will force the ugly "cost of production" dragon to surface again, and in only a matter of one or two years. Bimonthly issues ensure continued life for the longest period of time, more pages per issue and more time to prepare each issue as the pressure is lessened.

I wish there were a better answer, like more advertising revenue coming in or an increase in the subscriber base, but both are unlikely to happen. Please keep up the good work for as long as possible. It is appreciated more than you know.

Bill Gaskill
Grand Junction, Colorado

Saying goodbye

You wanted some feedback on raising the cost up to $50 for one year's subscription. You are correct, it would not go over at all. I don't know what to tell you to do. The problem is I plan to drop this magazine this year. I don't have any TI99/4A computer any more. I am now running on an IBM computer, but I do have PC99 from CaDD Electronics.

I do this with a heavy heart. Since 1986 I couldn't wait for my magazine to come in the mail. I have been with MICROpendium for the last 10 years. I also pick up all the back issues. This should tell you that you have done an excellent job in publishing this fine magazine. I will truly miss getting it in the mail, but I must move to the future, and the IBM is that future. The first computer I had was a Timex Sinclair. It just was not what I wanted. The TI99/4A turned out to be what I needed. I now have a Packard Bell 486 IBM compatible computer.

I wish you many years with MICROpendium. You have done an excellent job and you have a lot to be proud of. Thank you very much.

Tim Kirsch
Norfolk, Nebraska

Catch that 'Virus'

Just read a letter from Vern Jensen in the Lima Users Group newsletter. It seems Vern isn't getting a good response to his game offering, Virus Attack. I suppose I'm biased somewhat by the fact that I supplied some assembly help in the development of that game, but still I want to remind readers that this is a really *great* game for the TI.

Here at our house we have a Nintendo system on which we use the original Dr. Mario on which Vern based his game. When first I started corresponding with Vern, I was starting work on such a game myself in assembly language. Once I saw Vern's version, written in c99, I quickly abandoned that effort because I knew that even assembly wouldn't make a better game.

The action is fast and furious, but separated into levels with a pause between them, so one can get coffee or take care of any other needs between levels. The game can be played with or without joysticks. This games looks good, plays good, and can be a challenge for any TI game player. (I always play at the slow speed, because that's all I can handle.)

So I recommend to all you loyal readers that you get yourself a copy and send a donation of at least $15 to Vern Jensen, 910 Linda Vista Ave., Pasadena, CA 91103. It's well worth that much.

Bruce Harrison
Hyattsville, Maryland

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

BUGS AND BYTES

Delphi hosts conferences

In a "blast from the past" TI and Geneve conferences have taken place on Delphi, April 13 and 27. These events were at the instigation of Delphi TINET member Gary Fitzgerald who hopes for more to come.

Delphi to stay TI venue

At the April 27 conference, sysop Jerry Coffey says that the TINET on Delphi is "in for the long haul."

TINET, he says was "on indefinite hold while they worked up the WWW (World Wide Web) interface, but now the WWW is on indefinite hold."

Coffey speculates that they noticed that the rush to the Web was leaving a large niche market. (Delphi users can access Internet newsgroups with a TI or Geneve.)

THE ART OF ASSEMBLY — PART 58

The limits of an algorithm

By BRUCE HARRISON

This issue we're taking yet another look at the subject of random numbers. Yes, it seems we're doing a lot of that, but we want to be thorough, even if some of you might accuse us of "beating a dead horse." The subject today is some limitations we've found in our little random number algorithm that we've been using for some time. We found these limits mainly by doing some experiments with "nonsense" programs, three of which are provided in today's sidebar. These programs are not going to cause any big stir in the TI community, but will serve to illustrate our lesson.

We started this experimenting with a simple idea. Let's make a program that randomly puts the "@" character on screen until the screen is filled with them, then erase them in random fashion until the screen is cleared. That worked pretty much as expected, except that we noticed a tendency for certain columns to always get filled early in the process. To see better what was happening, we then changed our program so that instead of the "@" we'd use a randomly selected character in the range of 33 through 126 for each randomly chosen screen position. This makes a nice "mess" on the screen. We noticed, though, that on any particular run, we'd get only the odd numbered characters or only the even numbered ones. We could let that run continue as long as we wanted, and still see only odd or only even characters. Why? Starting a new run would change sometimes from odd to even, but we'd never see both odd and even in the same run!

THE EVEN DIVISOR PROBLEM

To examine this a bit more scientifically, we devised a different test program, (Part 1 of the sidebar) which just puts the random numbers themselves on the screen until we've filled it to the bottom, then clears the screen and starts over at the top. We put a delay into this program between each number's selection, so we'd have some time to look at the numbers before they'd disappear. What we found was that if we used an even number in R3 as the divisor, then the algorithm would alternate between odd and even numbers. Let's say, for example, we've used 100. The numbers produced would be "randomly" scattered in values between 0 and 99, but they would alternate between odd and even numbers no matter how long the run continued. If we changed the divisor to an odd number, (99, for example) the problem went away, and our random numbers would scatter nicely between 0 and 98, with no apparent pattern of odd and even numbers. Sometimes there would be three or four odds before an even, and vice versa, but there was no discernible pattern.

IT GETS WORSE

If the divisor is a small number, this gets worse. For example, using 2 simply makes alternating 1 and 0 values, since those are the only non-negative numbers less than 2. This, folks, is not random! Using four also creates a repeating non-random pattern that goes 3,0,1,2 over and over. This may start on any of those numbers, but the repeating is the same as long as it's left to run. Five, however, seems to be okay, in that the numbers 0 through 4 seem to come up in random order.

CAN WE FIX THIS?

Fortunately, there are solutions to this problem, and the sidebar shows a couple of them. In the source for the RANDEX program, you'll notice right after the LWPI WS a LI R3,4, which sets the range for random numbers to 0 through 3, but will create that repeating pattern. Just after that is a "commented out" line that says LI R3,5. A bit further down, just after label BUILD1, you'll see two commented out lines that say AI R5,-1 and JLT BUILD1. These lines are a solution to the problem! If you change the * to a space in each of those three lines, then assemble and run the program, you'll get the numbers from 0 through 3 that you were after, but they'll be random like they should be, not in a fixed pattern. What we've done is to use an odd number in R3, which gives us a number in R5 after BUILD1 that ranges from 0 through 4. We subtract 1 from that, so it ranges from -1 through 3, then if the result is less than zero, we simply repeat the BL @RANDNO at BUILD1. This will mean that, on average, one out of five numbers generated will be rejected, but that will not make any noticeable change in the speed of generating the correct numbers, and the sequence will be random, which is the desired result. This method can be applied for any situation in which the number in R3 would be even. You can use the next odd number, then subtract 1 from R5 after the BL @RANDNO, and repeat the BL if that's below zero.

For example, if the desired range is 0 through 99, you could do something like this:

```
LI  R3,101
GETRND BL  @RANDNO
AI  R5,-1
LT GETRND
```

That will yield nicely scattered numbers between 0 and 99 inclusive, but without the alternating odd-even problem. Of course if you don't care whether odd and even numbers alternate, you could just LI R3,100 and omit the AI R5,-1 and JLT lines.

TOSSING A FAIR COIN

We said earlier that the situation for 1 and 0 was particularly bad, in that loading R3 with 2 and then doing the BL @RANDNO would yield just alternating ones and zeros. True, but that too can be easily solved. Just do this:

```
LI  R3,5
BL  @RANDNO
ANDI R5,1
```

This will generate a random sequence of ones and zeros without any noticeable pattern. There will of course be cases where six or so ones appear in a row, and vice versa, but there will over time be as many ones as zeros, at least so far as human perception can tell. Any odd number greater than five can be used where we've shown LI R3,5. The result will be about the same. The computer will effectively toss a coin for you.

MORE THAN ONE WAY...

That usually ends "to skin a cat," but since we have cats here in

(See Page 10)

THE ART OF ASSEMBLY —

(Continued from Page 9)

the house, we dare not say that. Our cats might come up and skin a human. We're talking in this case about the second part of the sidebar, a nonsense program called "RANDSC"; this is the program that randomly scatters the characters from 33 through 126 about the screen at randomly chosen places, You'll notice that a few lines past label CKR1 there are two successive lines that BL @RANDNO. In the original program there was just one. Altogether, there are three BLs to RANDNO in the main loop. Originally, there were two, which meant that only the odd or even characters would show up in any given run. By adding a "dummy" BL line, we make the alternation case again, but we really don't care here because it's not important for the program's purpose.

WHY THE ALTERNATION?

It's built into the process by which the random number sequence is being generated. To prove this, we modified the program in PART 1 into the program in Part 3, called RAWEX. This one doesn't bother with a divisor, but just displays the "raw" random number that's in R5. These numbers will range from -32768 through 32767. Sure enough, when RAWEX runs, the numbers it generates will be randomly scattered over the range, but will alternate between odd and even, no matter how long the run. When we start a new run, the sequence may start with either an odd or an even number depending on the starting seed number we got when we "PRESS A KEY TO START," but they'll alternate between odd and even or vice versa from that point on. Thus, we have to do something about it if this alternation is not to be seen as a pattern.

THINGS THEY NEVER TAUGHT

Having been through many years of school, from first grade through a master's degree, including the bachelor of science in electrical engineering work, which included lots of math, your author might be expected to have a real grasp of mathematical "facts," but in the work leading to this column, we've uncovered a math fact that they never taught us.

It goes like this: If you divide a number by an even number, the remainder will take its even or odd quality from the number you started with. If you divide by an odd number, the remainder may be odd or even, not necessarily following the odd or even quality of the original number. Thus, when we use an even divisor in R3, the alternation between odd and even in the "raw" number will show up in the remainders. If an odd divisor is used, that alternation vanishes. Maybe that's not important except in this narrow field of work, but there it is anyway.

A FREE EXTRA 'GOODIE'

In part one of the sidebar, we've put in a new version of an old subroutine. This is called SHWINT, and uses an undocumented GPLLNK service to convert an integer at >835E to a string for display on the screen. The GPLLNK service (DATA >2F7C) treats the integer at >835E as an unsigned number, so the string will range from 0 through 65535. Here, however, we've added a few lines to the beginning so that it will recognize the "sign" bit, and display the negative numbers with a "-" sign. Thus, the positive numbers will show up as 0 through 32767, while the negative ones will show up as -1 through -32768.

This will work as shown so long as we're in the E/A Option 3 mode. If you're going to use it in another way, for example in an Option 5 program, you should use the Warren/Miller GPLLNK routine in the program instead of the REF GPLLNK that we've used here. In this particular program, the numbers to be displayed are all positive, but we left in the "negative handling" part just to provide our readers with another useful subroutine.

If you divide a number by an even number, the remainder will take its even or odd quality from the number you started with. If you divide by an odd number, the remainder may be odd or even, not necessarily following the quality of the original number.

There's one other thing you can do sometimes to improve the randomness of your numbers. In many programs, there's a key loop somewhere to wait for the user's input. If that key loop includes LIMI 2 and LIMI 0, as ours always do, you can put in an instruction like A @>8378,@>83C0 right after one of those BL @KEYLOO instruction, and that will adjust the seeding of the random number process each time that user input gets taken. The same can be done after a string or number input routine that u LIMI 2 and LIMI 0. Doing this will throw in another layer of unpredictability to your random number sequence, as nobody can know in advance what number will be at >8378 in such cases.

We're making no promises about the next column, except that it won't concern random numbers. Even your author can get tired of beating a dead horse.

SIDEBAR 58

```
      REF  VSBW,VMBW,KSCAN,GPLLNK
      DEF  START
STATUS EQU  >837C
START  LWPI WS           LOAD OUR WORKSPACE
       LI   R3,4         RANGE 0-3
       LI   R0,11*32+5   POINT ROW 12, COL 6
       LI   R1,PAK       "PRESS A KEY"
       LI   R2,20        20 CHARS
       BLWP @VMBW        WRITE THAT
SEED   MOVB @>83D7,R10   LOW BYTE TIMEOUT COUNTER
       BLWP @KSCAN       SCAN KEYBOARD
       LIMI 2            ALLOW INTERRUPTS
       LIMI 0            STOP THEM
       CB   @>837C,@ANYKEY KEY PRESSED?
       JNE  SEED         IF NOT, REPEAT
       A    @>8378,R10   ADD VERTICAL INT TIMER BYTE
       MOV  R10,@>83C0   PLACE R10 AT SEED
CLEAR  LI   R1,>2000     SPACE IN L.B. R1
       LI   R2,768       768 CHARS
       CLR  R0           SCREEN ORIGIN
CLRLP  BLWP @VSBW        WRITE ONE
       INC  R0           NEXT SPOT
```

(See Page 11)

THE ART OF ASSEMBLY —

(Continued from Page 10)

```
      DEC  R2          DEC COUNT
      JNE  CLRLP       RPT. IF NOT ZERO
      CLR  R0          SCREEN ORIGIN
BUILD CI   R0,>2FE     COMPARE TO 2 BYTES BEFORE END
      JLT  BUILD1      IF LESS, PROCEED
      JMP  CLEAR       ELSE RE-CLEAR SCREEN
BUILD1 BL  @RANDNO     GET A RANDOM NUMBER
      MOV  R5,@>835E   PLACE R5 AT >835E
      BL   @SHWINT     DISPLAY ON-SCREEN
      A    R2,R0       ADD LENGTH
      INC  R0          SKIP A SPACE
      MOV  R0,R1       PUT SCREEN ADDR IN R1
      ANDI R1,>001F    MASK OFF TO >1F
      CI   R1,>1E      NEAR END OF ROW?
      JLT  SCN         IF LESS, OKAY
      ANDI R0,>FFE0    MASK TO START OF ROW
      AI   R0,>20      MOVE DOWN ONE ROW
SCN   BLWP @KSCAN      SCAN KEYBOARD
      CB   @ANYKEY,@>837C KEY PRESSED?
      JEQ  EXIT        IF SO, EXIT
STDLY CLR  @>83D6      CLEAR TIMEOUT COUNTER
      LI   R4,12       12 = 6/60THS = 1/10 SECOND DELAY
DLY   LIMI 2           ALLOW INTS
      LIMI 0           STOP INTS
      C    @>83D6,R4   COMPARE TIMEOUT TO 1/10
      JLT  DLY         IF LESS, REPEAT
      JMP  BUILD       ELSE JUMP BACK
EXIT  LWPI >83E0       LOAD GPL WS
      B    @>6A        EXIT TO GPL INT.
RANDNO LI  R4,28645    BIG NUMBER IN R4
      MPY  @>83C0,R4   MULT. BY SEED
      AI   R5,31417    ADD BIG NUMBER
      MOV  R5,@>83C0   PUT BACK AT SEED
      CLR  R4          CLEAR HIGH WORD
      DIV  R3,R4       DIVIDE BY R3
      RT
SHWINT MOV @>835E,R2   GET NUMBER
      JEQ  SHWIN0      IF ZERO, JUMP
      JGT  SHWIN0      IF POSITIVE, JUMP
      NEG  R2          ELSE TAKE 2'S COMPLEMENT
      MOV  R2,@>835E   PLACE AT >835E
      LI   R1,>2D00    "-" IN L.B. R1
      BLWP @VSBW       DISPLAY -
      INC  R0          POINT TO NEXT SPOT
SHWIN0 CLR @STATUS     CLEAR GPL STATUS BYTE
      BLWP @GPLLNK     USE GPL LINK
      DATA >2F7C       CONVERT INTEGER TO STRING
      MOVB @>8361,R2   GET LENGTH
      SRL  R2,8        RIGHT JUST.
      MOVB @>8367,R1   GET L.B. ADDRESS
      SRL  R1,8        RIGHT JUST.
      AI   R1,>8300    ADD HIGH BYTE >83
      BLWP @VMBW       DISPLAY THE STRING
      RT               THEN RETURN
WS    BSS  32          OUR OWN WORKSPACE
PAK   TEXT 'PRESS A KEY TO START'
ANYKEY BYTE >20        HEX 20 FOR COMPARISON
      END
      REF  VSBW,VMBW,KSCAN
      DEF  START
START LWPI WS          LOAD OUR WORKSPACE
      LI   R0,11*32+5  ROW 12, COL 6
      LI   R1,PAK      "PRESS A KEY"
      LI   R2,20       20 CHARS
      BLWP @VMBW       WRITE
SEED  MOVB @>83D7,R10  L.B. SCRN TIMEOUT TO H.B. R10
      BLWP @KSCAN      SCAN KEYBOARD
      LIMI 2           INTS ON
      LIMI 0           INTS OFF
      CB   @>837C,@ANYKEY KEY PRESSED?
      JNE  SEED        RPT IF NOT
      A    @>8378,R10  ADD VERT TIMER TO R10
      MOV  R10,@>83C0  PLACE AT SEED
      LI   R1,>2040    SPACE AND "@" IN R1
BUILD CLR  R3          REG 3 = 0
      SWPB R1          SWAP BYTES IN R1
      LI   R9,BUFF2    POINT AT BUFFER 2
BRAN0 MOV  R3,*R9+     PLACE A WORD IN TABLE
      INC  R3          INC COUNT
      CI   R3,>300     COMPARE POINTER
      JLT  BRAN0       IF LESS, BACK
      MOV  R3,R6       PUT R3 INTO R6
      SLA  R6,1        DOUBLE THAT
BRAN1 BL   @RANDNO     GET A RANDOM NUMBER
      SLA  R5,1        DOUBLE REMAINDER
      MOV  @BUFF2(R5),R0   MOVE WORD FROM TABLE TO FAC
      BLWP @VSBW       WRITE A CHARACTER
      BLWP @KSCAN      SCAN KEYBOARD
      LIMI 2           INTS ON
      LIMI 0           INTS OFF
      CB   @ANYKEY,@>837C KEY?
      JNE  CKR1        JUMP IF NOT
      JMP  EXIT        ELSE EXIT
CKR1  CB   R1,@ANYKEY  SPACE IN L.B. R1?
      JEQ  DEC6        IF SO, JUMP
      MOV  R3,R9       SAVE R3
      MOV  R5,R10      SAVE CURRENT R5
      LI   R3,127-33   RANGE = 94
      BL   @RANDNO     GET RANDOM NUMBER
      BL   @RANDNO     TWICE
      AI   R5,33       ADD BOTTOM TO R5
      SWPB R5          SWAP
      MOVB R5,R1       PLACE BYTE IN L.B. R1
      MOV  R9,R3       GET OLD R3 BACK
      MOV  R10,R5      AND OLD R5 BACK
DEC6  DECT R6          SUBTR. 2 FROM R6
      MOV  @BUFF2(R6),@BUFF2(R5) REPLACE WORD JUST TAKEN
      DEC  R3          SUBTR 1 FROM R3
      JNE  BRAN1       IF NOT ZERO, REPEAT
      CB   R1,@ANYKEY  SPACE IN R1?
      JNE  STDLY       IF NOT, JUMP
SHWMSG MOV R1,R3       STASH R1 IN R3
      LI   R2,16       16 CHARS
      LI   R1,FINMSG   "FINISHED ..."
      LI   R0,11*32+8  ROW 12, COL 9
      BLWP @VMBW       WRITE MESSAGE
      MOV  R3,R1       GET OLD R1 BACK
STDLY CLR  @>83D6      CLEAR TIMEOUT
      LI   R4,240      TWO SECONDS DELAY
DLY   LIMI 2           INTS ON
      LIMI 0           INTS OFF
      C    @>83D6,R4   CHECK TIME
      JLT  DLY         IF LESS, REPEAT
      JMP  BUILD       ELSE TO BUILD
EXIT  LWPI >83E0       LOAD GPL WS
      B    @>6A        EXIT TO GPL INT.
RANDNO LI  R4,28645    BIG NUMBER IN R4
      MPY  @>83C0,R4   MULT. BY SEED
      AI   R5,31417    ADD BIG NUMBER
      MOV  R5,@>83C0   PUT BACK AT SEED
      CLR  R4          CLEAR HIGH WORD
      DIV  R3,R4       DIVIDE BY R3
      RT               THEN RETURN
WS    BSS  32          OUR OWN WORKSPACE
BUFF2 BSS  768*2       SELECTION LIST
```

(See Page 12)

THE ART OF ASSEMBLY —

(Continued from Page 11)

```
FINMSG TEXT 'FINISHED A CYCLE'
PAK    TEXT 'PRESS A KEY TO START'
ANYKEY BYTE >20          HEX 20 FOR COMPARISON
       END
       REF  VSBW,VMBW,KSCAN,GPLLNK
       DEF  START
STATUS EQU  >837C
START  LWPI WS           LOAD OUR WORKSPACE
       LI   R0,11*32+5   POINT ROW 12, COL 6
       LI   R1,PAK       "PRESS A KEY"
       LI   R2,20        20 CHARS
       BLWP @VMBW        WRITE THAT
SEED   MOVB @>83D7,R10   LOW BYTE TIMEOUT COUNTER
       BLWP @KSCAN       SCAN KEYBOARD
       LIMI 2            ALLOW INTERRUPTS
       LIMI 0            STOP THEM
       CB   @>837C,@ANYKEY KEY PRESSED?
       JNE  SEED         IF NOT, REPEAT
       A    @>8378,R10   ADD VERTICAL INT TIMER BYTE
       MOV  R10,@>83C0   PLACE R10 AT SEED
CLEAR  LI   R1,>2000     SPACE IN L.B. R1
       LI   R2,768       768 CHARS
       CLR  R0           SCREEN ORIGIN
CLRLP  BLWP @VSBW        WRITE ONE
       INC  R0           NEXT SPOT
       DEC  R2           DEC COUNT
       JNE  CLRLP        RPT. IF NOT ZERO
       CLR  R0           SCREEN ORIGIN
BUILD  CI   R0,>2FA      COMPARE TO 6 BYTES BEFORE END
       JLT  BUILD1       IF LESS, PROCEED
       JMP  CLEAR        ELSE RE-CLEAR SCREEN
BUILD1 BL   @RANDNO      GET A RANDOM NUMBER
       A    R2,R0        ADD LENGTH
       INC  R0           SKIP A SPACE
       MOV  R0,R1        PUT SCREEN ADDR IN R1
       ANDI R1,>001F     MASK OFF TO >1F
       CI   R1,>1A       NEAR END OF ROW?
       JLT  SCN          IF LESS, OKAY
       ANDI R0,>FFE0     MASK TO START OF ROW
       AI   R0,>20       MOVE DOWN ONE ROW
SCN    BLWP @KSCAN       SCAN KEYBOARD
       CB   @ANYKEY,@>837C KEY PRESSED?
       JEQ  EXIT         IF SO, EXIT
STDLY  CLR  @>83D6       CLEAR TIMEOUT COUNTER
       LI   R4,28        14/60THS DELAY
DLY    LIMI 2            ALLOW INTS
       LIMI 0            STOP INTS
       C    @>83D6,R4    COMPARE TIMEOUT TO 1/10
       JLT  DLY          IF LESS, REPEAT
       JMP  BUILD        ELSE JUMP BACK
EXIT   LWPI >83E0        LOAD GPL WS
       B    @>6A         EXIT TO GPL INT.
RANDNO LI   R4,28645     BIG NUMBER IN R4
       MPY  @>83C0,R4    MULT. BY SEED
       AI   R5,31417     ADD BIG NUMBER
       MOV  R5,@>83C0    PUT BACK AT SEED
SHWINT MOV  R5,@>835E    GET NUMBER
       JEQ  SHWIN0       IF ZERO, JUMP
       JGT  SHWIN0       IF POSITIVE, JUMP
       NEG  R5           ELSE TAKE 2'S COMPLEMENT
       MOV  R5,@>835E    PLACE AT >835E
       LI   R1,>2D00     "-" IN L.B. R1
       BLWP @VSBW        DISPLAY -
       INC  R0           POINT TO NEXT SPOT
SHWIN0 CLR  @STATUS      CLEAR GPL STATUS BYTE
       BLWP @GPLLNK      USE GPL LINK
       DATA >2F7C        CONVERT INTEGER TO STRING
       MOVB @>8361,R2    GET LENGTH
       SRL  R2,8         RIGHT JUST.
       MOVB @>8367,R1    GET L.B. ADDRESS
       SRL  R1,8         RIGHT JUST.
       AI   R1,>8300     ADD HIGH BYTE >83
       BLWP @VMBW        DISPLAY THE STRING
       RT                THEN RETURN

WS     BSS  32           OUR OWN WORKSPACE
PAK    TEXT 'PRESS A KEY TO START'
ANYKEY BYTE >20          HEX 20 FOR COMPARISON
       END
```

TI Casino to be updated

TI Casino is currently being updated to 16 game modules, according to a press release from Notung Software. The new Casino will include the original programs; Blackjack, Video Poker, Acey-Ducey (revised — not as easy to win), Keno, Baccarat, Roulette and Craps (revised — now with free odds), plus the Casino Supplement package; Faro and Club Notung (revised — now with multiple shows), according to the manufacturer. In addition, the manufacturer plans to add the following game modules:

- Single-Deck Blackjack
- Paigow Poker — the user against the dealer
- 5 Card Stud — the user against three computer-generated players.
- Wheel of Fortune
- Casino Solitaire — totally joystick operated
- Thoroughbred Racing — with a 240-horse database, printable Daily.

The manufacturer plans not to issue Casino any longer in SSSD format. It will come on two DSSD disks or one DSDD disk. It will also be available in PC99 and V9T9 formats.

Weekly progress reports are available on the Internet at Notung's home page: http://ourworld.compuserve.com/homepages/notung/

All products formerly available from now defunct MS Express are now available through Notung. This includes Page Pro Cataloger ($20), Genealogy Plus ($20) and TI99/4A Software Database ($20). The one-page illustrated Notung Catalog is available free by writing c/o Notung Software, 7647 McGroarty Street, Tujunga, CA 91042 or e-mailing : 102277.3452@compuserve.com.

TUTORIAL

Using the Hewlett-Packard printer command language with a TI

By BOB GROSSART

This is reprinted from TIC TOC, *the newsletter of the Rocky Mountain 99ers—Ed.*

E_c(s0p10h6v0s3b3T

Introduction. Now that I have your attention, we can look into the world of the Hewlett-Packard (HP) Printer Command Language, known as HP PCL. I became interested in the HP PCL while trying to adapt TI99/4A programs to print with my recently acquired HP DeskJet 310 printer. Initially, ASCII files printed OK (sort of). But it became obvious that HP was far from being Epson-compatible. I needed to learn what I could about my HP printer if I wanted it to be useful with my TI. From HP, I ordered a Technical Reference Manual for my HP310. This 224-page manual contained very detailed information regarding printer commands and use of the PCL, in my case PCL-3. It was well worth the $5.25 plus tax, sent FEX. Listed at the end of this article are this manual and two other books that have useful information on both the DeskJet and LaserJet HP printers.

HP PCL Architecture. The HP PCL provides application programs a way to control HP device features using a common printer language and architecture.

PCL is essentially a collection of commands, known as control and escape sequence codes, to control printer operations. HP categorizes general types of device functions into levels. Several levels of PCL have been implemented into different HP devices, but they are all backward compatible. PCL-4 incorporates all of PCL1, 2 and 3. Also, all levels of PCL ignore PCL commands they do not support. The levels are:

PCL-1: :Print and Space PCL. Supports standard codes (e.g. CR, LF, underline, basic graphics)

PCP-2: EDP/Transaction PCL. For printers designed for transition processing. Adds margin, line spacing, page length, pitch, others.

PCL-3: Office Word Processing PCL. For printers designed for word processing. Adds commands to support LaserJet printers.

PCL-4: Page Formatting PCL. Came into use with the LaserJet Plus printers.

PCL-5: Enhanced but more complex PCL to support the LazerJet 111 family of printers

The HP PCL architecture consists of two major elements: PCL Kernel and Extensions to the Kernel. Extensions provide features that are device-specific (e.g., print color, envelope printing). The Kernel consists of the following feature groups: Device Control, Movement, Page Presentation, Fonts and Rendering(or graphics). My Technical Reference Guide describes each if these features in detail.

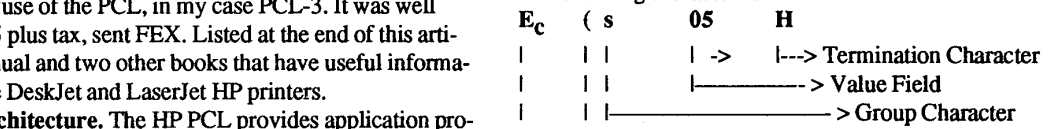
PCL Codes. There are two general types of PCL commands: Control and Escape Sequences (sounds familiar so far).

Control Codes. Beginning with CTRL (IBM) or CTRL-U (TI), a single character (ASCII table in range of 0 through 32 decimal) that initiates a printer function (e.g., Carriage Return. Line Feed).The codes for my printer are listed in Table 111, which correspond to the special character mode commands in the TIW manual on page 146. BASIC uses the CHR$ function. e.g. CHR$(12) results in Form Feed.

Escape Sequences. Two or more characters with the first character always ASCII 27 decimal or 1B hexadecimal (identified as E_C).

Two Character Escape Sequence. E_c is followed by any character from ASCII table within range of 48-126 decimal, e.g., E_cE resets the printer

Parameterized Escape Sequence. The escape sequence has the following characteristics:



Parameterization Char.: Within range of 33-47 decimal. Indicates that this escape sequence is parameterized.

Group Char.: Within range of 96-126 decimal (' through /). Specifies group type.

Value Field: Within range of 48-57 decimal (0 through 9). May be preceded by + or - and contain digits after a decimal point. Range of -32767 to 32767. A value of 0 if not specified and is required.

Termination Char.: Within range of 64-94 decimal (@ through ^). Specifies parameter for value held. Terminates escape sequence when a capital letter.

Binary Data: Not required for example code above, but is eight-bit data (e.g., graphics). The Value Field specifies the number of bytes of binary data to immediately follow escape sequence.

In accordance with the PCL-3 command list, the example escape sequence above will produce a font pitch selection of five characters per inch. This assumes the printer has the built-in font with this characteristic/ My printer has 37 different attribute categories controlled by escape codes.

Combined Escape Sequences: More than one escape sequence may be combined into one escape sequence using the following rules:

1. The first two characters following the "E_c" must be the same in all commands combined.

2. All alphabetic characters within the combined command are lowercase, except the final termination character.

(See Page 14)

HP TUTORIAL —

(Continued from Page 13)

3. Commands are executed in order, left to right.

For example, the combined sequence:

E_C(s0pl0h6v0s3b3T

is Font No. 56 on my printer which is specified by the attributes of Fixed Spacing, Pitch 10, Point Size 6, Upright Style, Bold Weight and Courier Typeface. This is equivalent to the following individual escape sequences:

E_C(s0P	Fixed Spacing
E_C(s10H	Pitch 10
E_C(s6V	Point Size 6
E_C(s0S	Upright
E_C(s3B	Bold Weight
E_C(s3T	Courier

HP cautions that the escape sequence should not have any spaces. My manual gives not only the alphanumeric codes, but also the corresponding decimal and hexadecimal equivalents. Table I contains some of the more common word processing commands. A full list is in the HP Technical Reference Manual, including graphics, which I have yet to explore. Table II is an index of Escape Sequence feature groups.

Application to TI99/4A: Having a list of PCL codes, the next step was to incorporate them into TI programs. I have tried several combinations of decimal, hex, CHR$ and CTRL-U commands. So far my success has been limited.

Word Processing by TIW/FW:

a. *The CTRL-U* trick works as follows:

1. Bold print is desired
2. PCL-3 command E_C(s3B
3. Embed in text where required, the following keyboard sequence(no spaces):

CTRL-U FUNC-R CTRL-U (s3B

4. To return to normal stroke weight embed the following code:

CTRL-U FUNC-R CTRL-U (s0B

(Remember all commands remain in effect until canceled by another command, send printer reset or cycle printer power off/on.)

b. *Transliterate Commands* work as usual with the formatter.

.TL 35:27,40,115,53,72
.TL 33:27,40,115,49,48,72
#TEST HP PRINTER!

would print at five characters per inch. The equivalent PCL-3 commands are:

E_C(s5H replacing # and E_C(s10H replacing !

BASIC Program. The following program was used to test printer with TI99/4A, with result shown.

```
10 OPEN #1:"PIO"
20 PRINT#1:CHR$(27)&"(sSH"
30 PRINT#1: "THIS IS A TEST
":"OF THE HP PRINTER"
40 PRINT#1:CHR$(27)&"(s10H"
50 PRINT#1:"THIS COMPLETES
1st HP PRINTER TEST"
60 PRINT#1: CHR$(27)&"(s511
"Test HP again"
70 PRINT#1:CHR$(27)&"(s10H"
&"This completes test"
80 CLOSE#1
```

```
THIS IS A TEST
OF THE HP PRINTER

THIS COMPLETES 1st HP PRINTER TEST
Test HP again
This completes test
```

Summary: Although I have not addressed graphics modes or extended programming methods (my lack of understanding), I hope this information may be of some interest if you acquire an HP printer for use with your TI99/4A.

References

1. Hewlett-Packard Company, "HP DeskJet 310 Printer, Technical Reference Guide," Manual Part Number C2621-90172.

2. Crane, Mark W., Pierce, Joseph R., with Holzgang, Danial A., "LaserJet Companion," Microsoft Press.

3. Cummings, Steve, "DeskJet Unlimited," Peachpit Press Inc.

Table I. HP PCL-3 ESCAPE SEQUENCES

Code Name	Printer Feature	Command	Decimal Equivalent
Reset		EcE	027 069
Self Test		Ecz	027 122
Point Size	# 1/72 inch	Ec(s#V	027 040 115 #..# 086
Print Pitch	Char./Inch	Ec(s#H	027 040 115 #..# 072
Stroke Weight	Normal	Ec(s0B	027 040 115 048 066
" "	Bold	Ec(s3B	027 040 115 051 066
Style	Upright	Ec(s0S	027 040 115 048 083
"	Italic	Ec(s1S	027 040 115 049 083
Spacing	Proportional	Ec(s1P	027 040 115 049 080
"	Fixed	Ec(s0P	027 040 115 048 080
Print Quality	Letter	Ec(s2Q	027 040 115 050 081
" "	Draft	Ec(s1Q	027 040 115 049 081
Underline	Single Fixed*	Ec&d1D	027 038 100 049 086
	*other modes available		
Underline Off		E_C&d@	027 038 100 064
Line Spacing	Lines/Inch*	E_C&*l*#D	027 038 108 #..# 068

*E_C&*l*8D is 8 lines per inch. Also, the *l* is a *lower case* "l" or "ell" but this shows up better. The HP manual uses a lower case *script* "ell," also to distinguish from a "one"

Vertical Pos.	Decipoints*	E_C&a#V	027 038 097 #..# 086

*Increments of 1/720 inch. A +240 is one-third inch down and a -240 is one-third inch up, for value of #.

(See Page 15)

HP TUTORIAL —

(Continued from Page 14)

Note: In the above list the symbol # represents a decimal number, while #..# represents a decimal number represented as ASCII code (10 is 49 48).

All upper/lower case values and no spaces must be followed.

TABLE II. ESCAPE SEQUENCE INDEX

(Spaces shown for clarity)

Escape Sequence	*Description*
E_c & *l* # A	Paper Size
E_c & *l* # C	Vertical Motion Index
E_c & *l* # D	Line Spacing
E_c & *l* # E	Top Margin
E_c & *l* # F	Text Length
E_c & *l* # H	Media Source
E_c & *l* # L	Perforation Skip Mode
E_c & *l* # O	Logical Page Orientation
E_c & *l* # P	Page Length
E_c & a # C	Horizontal Position, Columns
E_c & a # H	Horizontal Position, Decipoints
E_c & a # L	Left Margin
E_c & a # M	Right Margin
& a # R	Vertical Position, Rows
E_c & a # V	Vertical Position, Decipoints
E_c & b # T	Dry Time
E_c & d # D	Automatic Underlining ON
E_c & d @	Automatic Underlining OFF
E_c & k # G	Line Termination
E_c & k # H	Horizontal Motion Index
E_c & k # W	Text Printing Mode
E_c & p # X	Transparent Print Mode
E_c & s # C	End-of-line-Wrap
E_c ([ID]	Character Set, Primary
E_c (s # B	Font Stroke Weight, Primary
E_c (s # H	Font Pitch, Primary
E_c (s # P	Font Spacing, Primary
E_c (s # Q	Font Quality, Primary
E_c (s # S	Font Style, Primary
E_c (s # T	Font Typeface, Primary
E_c (s b # V	Font Height, Primary
E_c) [ID]	Character Set, Secondary
) s # B	Font Stroke Weight, Secondary
E_c) s # H	Font Pitch, Secondary
E_c) s # P	Font Spacing, Secondary
E_c) s # Q	Font Quality, Secondary
E_c) s # S	Font Style, Secondary
E_c) s # T	Font Typeface, Secondary
E_c) s b # V	Font Height, Secondary
E_c * b # M	Raster Graphics Compressor
E_c * b # S	Seed Row Source
E_c * b # V	Transfer Graphics Plane 1
E_c * b # W	Transfer Graphics Row 1
E_c * b # Y	Relative Vertical Pixel Movement
E_c * o # D	Raster Graphics Depletion
E_c * o # Q	Raster Graphics Shingling
E_c * p # X	Horizontal Position, Dots
E_c * P # Y	Vertical Position, Dots
E_c * r # A	Start Raster Graphics
E_c * r # Q	Raster Graphics Quality
E_c * r # S	Raster Graphics Width
E_c * r # U	Number of Graphics Planes
E_c * r b C	End Raster Graphics
E_c * t # R	Raster Graphics Resolution
E_c 9	Clear Side Margins
E_c E	Reset Printer
E_c Y	Display Functions Mode ON
E_c Z	Display Functions Mode OFF
E_c z	Printer Self Test

TABLE III. HP PCL-3 CONTROL CODES

(CTRL-U Special Character Mode on TI99/4A)

	TI		IBM
Code Name	**Key Press**	**Key Press**	**Decimal**
Backspace	SHIFT H	CTRL H	08
Horizontal Tab	SHIFT I	CTRL I	09
Line Feed	SHIFT J	CTRL J	10
Form Feed	SHIFT L	CTRL L	12
Carriage Return	SHIFT M	CTRL M	13
Shift Out	SHIFT N	CTRL N	14
Shift In	SHIFT O	CTRL O	15
Escape	FCTN R	CTRL R	27
Space			32

Portland user group gets e-mail address

The Portland Users of Ninety-Nines in Portland, Oregon, have a new e-mail address. It is punn@worldnet.att.net.

Hidden characters

Using invisible characters in filenames

By STEVE PATTERSON

Patterson is listed as a member of New Horizons. MICROpendium picked up this article from the Spirit of 99 Newsletter *of the Central Ohio Ninety-Niners. The newsletter thanks the TEXPAC BBS of TIsHUG.*

Hidden characters is a term that I have given for the use of characters in a filename that are unable to be seen on most disk catalogs. These types of characters can be used in both programs and files but there is a huge difference. With files, if you copy the program with an Extended BASIC file copier such as my "Copier" then you can use any character ever imagined.

For program type HC I have only found one to this day, which is ASCII 127 or FCTN-V. This is the only character that is hidden and has a key-in. You can use this character at the end of a filename to make it hard for others to run the program because they cannot figure out how to spell the filename because they do not notice the last hidden character. Of course you can use more than one and they do not have to be at the end. You could put one in the middle of the filename or you could let the entire filename be character 127. But having it at the end makes it a lot less noticeable.

Now how do you use ASCII 127 and possible ASCII 1-29 in a filename for a file? This is where you will have to have some XBASIC programming knowledge. You have to use either my program mentioned above or one of your own which you can write from scratch. Basically how to get any character at the end of a filename is to open that file in XB, then open a new file with a couple of hidden characters at the end of the filename. Then read each record of the file and save it to the new file.

> You would want to place hidden characters only on certain files that only you want to see and that will not be updated often.

So you have actually copied the file itself and in the process you have added several invisible or hidden characters at the end of the filename. With the possibility of any char from 1 to 29, only you know the characters at the end. So, say you have a filename like: "DSK1.LIST"; you can change that normal and easy-to-read file into the almost impossible-to-open file: "DISK1.LIST"&CHR$(2)&CHR$(25). This portion will place an ASCII Char 2 right after the "T" in the name. It will not be seen on most catalogs but you will know what it is because you wrote the two chars down somewhere safe.

One bad thing about this is that later you cannot load this file into any program without changing it back to the norma way or by going in and changing the ope statements in the program so that the hid den characters are in the filename. So yo would not want to place hidden characte in a filename that is updated every day Only on certain files that you only want t see and that will not be updated ofte would this be of any use.

The reason I keep saying that most cata logers will not be able to read the end char acters is because I wrote one that can de tect when there is a character in the file name that you would not normally see. Fo those who are not clear on the type of pro gram that will add these hidden characte to a filename, read on.

```
100 OPEN #1:"DSK1.LIST",INPU
T,DISPLAY,VARIABLE 80
110 OPEN #2:"DSK1.LIST"&CHR$
(4)&CHR$(16)&CHR$(23),OUTPUT
,DISPLAY,VARIABLE 80
120 IF EOF(1) THEN 160
130 LINPUT #1:A$
140 PRINT #2:A$
150 GOTO 120
160 CLOSE #1
170 CLOSE #2
180 END
```

The program above will give you a copy of the file LIST with three hidde characters at the end of the file, these char acters being ASCII 4,16,23. Hope you fin this new and experimental process of lock ing up files of some use and of some hel in keeping away the unwanted and un needed.

More vendors announced for fair

Glenn Bernasek of TI-Chips has updated the list of vendors who have committed for the Multi User Group scheduled May 25 in Brookpark, Ohio. TI-Chips and the Northcoast 99ers (Cleveland area TI users groups) will sponsor the event.

Vendors signed up include Tim Tesch, S&T Software; Bruce Harrison, Harrison Software; Mickey Cendrowski, Notung Software and West Penn 99ers; Jim Krych, SouthWest 99ers (Super AMS); Ron Markus, Ramcharged Computers; Mike Wright, CaDD Electronics; Charles Good, Lima Users Group; and Don Walden, Cecure Electronics and Milwaukee Users Group.

For further information, contact Bernasek at 13246 Harper Rd. Strongsville, OH 44136 or call (after 9 p.m. Eastern time) at (216) 846-0865 or Internet dd314@cleveland.freenet.edu.

Booting your Geneve from a Horizon RAMdisk

By JACK MATHIS

This is reprinted from the January 1995 newsletter of the SouthWest Ninety-Niners.—Ed.

So you say you've tried FORM3MEG by Jim Schroeder and you still cannot "boot" your Geneve! You can get your Horizon to format at the proper amount of "sectors" but you cannot get it to load the "SYSTEM/SYS" file? You can "look" at the disk with "Disk Utilities" or some other sector editor and "see" the file is there, but it won't "auto-load"? Well, you weren't alone!

I tried to format my HRD 4000 and Mike Doane's HRD 3000 and although they formatted to the correct size, they would not load the SYSTEM/SYS or autoload the AUTOEXEC file. I examined the HRDs (using the late John Birdwell's Disk Utilities) and made an interesting discovery.

The "FORM3MEG" program does indeed "mark" the HRD correctly as per the correct number of sectors, but the Geneve cannot locate the file. The problem is when the Geneve is attempting to boot up the file the Geneve is unaware that *every* file's "starting" sector has to be multiplied depending on the size of the HRD.

The "fix" for this was simple after I discovered the problem.

Physically set the Horizon CRU address to 140-0 with the dipswitches locating on the HRD. You can find the settings in your HRD manual.

Use FORM3MEG to format your HRD at CRU 1401, set as boot RAMdisk and follow the instructions to load SYSTEM/SYS. The RAMdisk is now set as DSK8.

Using Disk Utilities, go to File Utilities; select the File Report option; type SYSTEM/SYS at the prompt "File Name" (you *must* use capital letters!) and "DSK8" at the drive prompt.

Look for the sector where the FDR is located. Go to the "sector utilities" portion of the program.

Edit the FDR sector — Change byte 28 (>1C) in Hex Mode to read 00F21D.

Your system should boot from your Horizon now.

Now to make SYSTEM/SYS file boot your AUTOEXEC.

Go back to the "File Utilities" on the "SYSTEM/SYS" file and using the "replace string" option search for the following Hex code and replace the 1400 with 1401 (sets DSK6 to CRU 1401 — REMAP 6N). If you have a second HRD "addressed" at 1700, then changing the 1600 with 1701 (sets DSK7 to CRU 1701 — REMAP 7Q) allows you to access it as DSK7.

020C 1400 1014 020C 1600 1011 020C 1401 100E 020C
1601

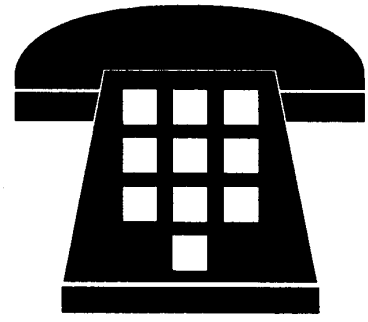
Next, search for "DSK1.AUTOEXEC" and change it to "DSK6.AUTOEXEC"; for automatic ASSIGN of drives, search for:

"DSK1" is the "A" drive
"DSK2" is the "B" drive
"DSK3" is the "C" drive — I changed this to "HDS1"
"DSK4" is the "D" drive — I changed this to "HDS2"
(Do *not* change these drive numbers if you have more than two floppy drives.) This is equivalent to "ASSIGN C=HDS1:" in the "autoexec" file.
"DSK5" is the "E" drive — RAMdisk if called for in AUTOEXEC
"HDS1" is the "F" drive — I changed this to "DSK6"
"HDS2" is the "G" drive — I changed this to "DSK7"
"HDS3" is the "H" drive — I changed this to "DSK8"

This will enable you to boot from your HRD. This modification has worked on HRD 3000s and 4000s.

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

FAIRYWALTZ, L'EU VIVE and ROOTS

Waltzing your way to user-defined subprograms

By LUCIE DORAIS
©1996 L. Dorais

Shall we dance? Or simple Waltz choreography, to show off the user-defined subprograms.

The waltz is a three-time rhythm, often played as a simple melody in Treble Clef, and an accompaniment in the Bass Clef. Most often, the chord in the accompaniment is broken into the tonic note played alone, then the two other notes of the chord played together twice, for the characteristic "oom-pah-pah" sound. And all bars follow the same pattern, at least in the easiest arrangements, which is what I have used here.

For the choreography, we have two couples that will whirl and spin around the screen, following the beat. To make the system "universal", i.e. easy to apply to any simple waltz, I have used a SUB BAR that is called once for each bar in the music. The subprogram plays the notes and move the two sprite couples. To make it easy to adapt to other songs, I have also done the initialization part as a sub START, called only once.

SUB START sets text colors to white on black, displays a centered title, and then defines three sprites, two being a mirror of each other, i.e. a couple with the man facing left and another with him facing right, char. 128-131 and 132-135. The third sprite, for characters 136-140, is the spinning couple seen from the back. As it is shown in silhouette (sprites can have only one color), the direction of their spin is irrelevant: both times, you see only the silhouette of the woman's crinoline.

I am grateful to Jeremy Frank, from Windsor and Ottawa, for his Makechar program, part of his fairware disk distributed by our library. Not only did it make it easy for me to mirror my first sprite and design the third one by slightly modifying the first one, but it also allowed me to check the animation before committing everything to the program. I also used my own Sprite Tester to find the starting positions of my two sprites. End of the commercial.

> Even if the sub is complicated, with all its parameters, it saves a lot of code. Just imagine lines 10080-90 repeated 20 times! And, because most of the values passed are different from one bar to the next, a GOSUB solution was not practical.

The starting sprites are called in line 10050 and are made as big as possible by a magnifying factor of 4. One is pale blue (8), the other pink (10). The third and fourth parameter for each sprite is its starting position: dotrow 81, dotcolumns 49 and 193, respectively, passed to the sub as parameters in the CALL.

Lines 140-150 define the variables used for the notes, from low D to high-high A (above the Treble Clef); the rest R is 40000, and finally we define the row velocity and column velocity of our sprites as RV and CV. This abundance of variables is placed before the Pre-Scan, so the latter, in line 160, deals only with those variables not yet defined. All the CALL statements are used mostly in the user-defined subs, so there's no need to Pre-Scan them, since the subs are outside the prescanning range (line 9999).

The score of our waltz becomes a series of CALL BARs, which carry over a list of ten parameters: row and column velocity for each sprite, then the three notes in the Treble Clef (the same note is repeated t allow for half notes and dotted half note in the score), and finally the three notes fo the Bass Clef broken chords. To make th typing job easier, use the REDO functio in command mode: type line 180, the pres FCTN-8: the line is brought again at th bottom on the screen; type over the new line number, and type over the parameter that are different. Be careful with the "+" signs in the choreography, which give op posite movements to our couples.

Even if the sub is complicated, with al its parameters, it saves a lot of code. Jus imagine lines 10080-90 repeated 20 times And, because most of the values passe are different from one bar to the next, a GOSUB solution was not practical.

Since the score has a different ending for the two times the melody is played, w use an ON GOTO statement in line 300 control the flow of the programmed music Line 390 simply brings our couples to a much needed rest while Tex asks you if you want to dance again.

Now to SUB BAR, lines 10060-10090 our ten parameters are listed in brackets as R and C velocities for each sprite, then the three Treble Clef notes and the three Bass Clef notes. Line 10070 defines the contrasting patterns for each sprite. P2 is set to 132 each time P1 is 128, and to 128 each time P1 is not 128, including the first time the sub is called, when P1 is still zero. Then P1 is defined the same way: when P2 is 132, P1 will be 128, and vice-versa. The relational expressions in brackets return a "0" if they are false, and "-1" if true. This line uses the notion that the variables in a user-defined sub keep their values from one CALL to the next.

Line 10080 changes the patterns of the sprites before playing the first beat of the bar: one note T1 in the melody, one (B1, the chord tonic note) in the accompaniment. The duration of this first beat ha been set to 300. Line 10090 plays the second beat, with a slightly longer duration,

(See Page 19)

FAIRYWALTZ —

(Continued from Page 18)

as in a true Viennese waltz. The second note of the melody bar and the two upper notes of the Bass chord are played. The sprite pattern is then changed to the "back-view" sprite of character 136. Then the third and final beat is played — third melody note, and same two notes as previous beat for the accompaniment (B2 and B3). Before leaving the SUB, P1 is set to P2 so that next time the sub is called, we will get a reverse sprite pattern from the previous CALLed BAR.

To show that this pattern could be used with any waltz, I picked another simple score and, using the Redo function (and the Erase to delete the lines that I didn't need, since this second score is much shorter), I quickly modified the score portion of FAIRYWALTZ (lines 100-400) for the second tune, a French song, L'EAU VIVE.

Besides changing the parameters for each CALL BAR, I modified lines 140 and 150 to use only the notes I needed, and the row and column velocities to meet my new choreography requirements. Line 125 has been added to display the apostrophe in the title in the same color as the other characters. The rest of the program is exactly the same as the previous waltz, with one exception — since each song has a different feeling, I changed the durations of the notes, from 300-450-300 to 250-350-250, in the three CALL SOUNDs (lines 10080-10090).

If you want to use this template for other waltzes, you could save lines 9999-10090 as a MERGEable file (this is why I used higher line numbers), remembering to adapt the sound durations to what you want. You could then modify it a bit for a four-beat rhythm. How about a rock beat? but make sure to change the costume of our lady!

FAIRY WALTZ

```
100 ! ** FAIRY WALTZ ** by L
. Streabogg (1835-1886) !081
110 ! progr. by L.Dorais/Ott
awa UG/Sept. 1989 !065
120 !!131
130 CALL CLEAR :: CALL START
("LITTLE FAIRY WALTZ",81,49,
81,193)!125
140 LD=147 :: LFS=185 :: LG=
196 :: A=220 :: B=247 :: C=2
62 :: D=294 :: FS=370 :: G=3
92 :: HA=440 :: HAS=466 :: H
B=494 !056
150 HC=523 :: HD=587 :: HE=6
59 :: HFS=740 :: HG=784 :: H
HA=880 :: R=40000 :: RV=8 ::
 CV=13 !168
160 GOTO 170 :: A$,TIME :: !
@P- !132
170 FOR TIME=1 TO 2 !031
180 CALL BAR(RV,CV,-RV,-CV,H
B,HB,HC,LG,D,B)!179
190 CALL BAR(0,CV,0,-CV,HD,H
D,G,LG,D,B)!083
200 CALL BAR(-RV,CV,RV,-CV,F
S,FS,FS,A,D,C)!142
210 CALL BAR(-RV,0,RV,0,FS,R
,R,A,D,C)!254
220 CALL BAR(-RV,-CV,RV,CV,H
C,HC,HD,LD,D,C)!180
230 CALL BAR(0,-CV,0,CV,HE,H
E,HA,LD,D,C)!149
240 CALL BAR(RV,-CV,-RV,CV,G
,G,G,LG,D,B)!233
250 CALL BAR(RV,0,-RV,0,G,R,
R,LG,D,B)!253
260 CALL BAR(-RV,CV,RV,-CV,H
G,HG,HFS,LG,D,B)!019
270 CALL BAR(0,CV,0,-CV,HHA,
HHA,HG,LG,D,B)!037
280 CALL BAR(RV,CV,-RV,-CV,H
G,HG,HG,LD,C,A)!188
290 CALL BAR(RV,0,-RV,0,HFS,
R,HA,LD,C,A)!201
300 ON TIME GOTO 310,350 !04
3
310 CALL BAR(RV,-CV,-RV,CV,H
FS,HFS,HE,LD,C,LFS)!002
320 CALL BAR(0,-CV,0,CV,HA,H
A,HAS,LD,C,LFS)!129
330 CALL BAR(-RV,-CV,RV,CV,H
B,HB,HB,LG,D,B)!178
340 CALL BAR(-RV,0,RV,0,HB,R
,R,LG,D,B):: GOTO 400 !162
350 CALL BAR(-RV,-CV,RV,CV,H
E,HE,HE,LD,C,LFS)!090
360 CALL BAR(0,-CV,0,CV,HA,H
A,HB,LD,C,LFS)!047
370 CALL BAR(-RV,-CV,RV,CV,G
,G,G,LG,D,B)!233
380 CALL BAR(RV,0,-RV,0,G,R,
R,LG,R,R)!027
390 CALL PATTERN(#1,132,#2,1
28):: CALL MOTION(#1,0,0,#2,
0,0)!072
400 NEXT TIME !197
410 DISPLAY AT(24,11):"AGAIN
? Y" :: ACCEPT AT(24,18)SIZE
(-1)VALIDATE("YN"):A$ !076
420 IF A$="N" THEN END ELSE
DISPLAY AT(24,11):"" :: GOTO
 170 !126
9999 !@P+ !062
10000 SUB START(A$,R1,C1,R2,
C2):: CALL SCREEN(2):: FOR X
=4 TO 8 :: CALL COLOR(X,16,2
):: NEXT X !164
10010 X=(28-LEN(A$))/2+.5 ::
 DISPLAY AT(1,X):A$ !109
10020 CALL CHAR(128,"070F0A0
B0B0307070F0F1F1F3F3F3F021C1
C08FC1CFC1C1C9C88C8C8E8E8E81
8")!158
10030 CALL CHAR(132,"3838103
F383F383839111313171717 18E0F
050D0D0C0E0E0F0F0F8F8FCFCFC4
0")!044
10040 CALL CHAR(136,"0303010
30303030307070F0F1F1F1F02808
0008080808080C0C0E0E0F0F0F08
0")!124
10050 CALL MAGNIFY(4):: CALL
 SPRITE(#1,128,8,81,49,#2,13
2,10,81,193):: SUBEND !110
10060 SUB BAR(R1,C1,R2,C2,T1
,T2,T3,B1,B2,B3)!215
10070 P2=128-4*(P1=128):: P1
=128-4*(P2=128)!246
10080 CALL PATTERN(#1,P1,#2,
P2):: CALL MOTION(#1,R1,C1,#
2,R2,C2):: CALL SOUND(300,T1
,5,B1,10)!189
10090 CALL SOUND(450,T2,5,B2
,10,B3,10):: CALL PATTERN(#1
,136,#2,136):: CALL SOUND(30
0,T3,5,B2,10,B3,10):: P1=P2
:: SUBEND !167
```

L'EAU VIVE.

```
100 ! ** L'EAU VIVE ** by Gu
y Beart !145
110 ! progr. by L.Dorais/Ott
awa UG/Sept. 1989 !065
120 !!131
125 CALL CHARPAT(39,A$):: CA
```

(See Page 20)

FAIRYWALTZ —

(Continued from Page 19)

```
LL CHAR(61,A$)!233
130 CALL CLEAR :: CALL START
("L=EAU VIVE",106,33,106,209
)!109
140 LB=123 :: LC=131 :: LE=1
65 :: LF=175 :: LG=196 :: C=
262 :: D=294 :: E=330 :: F=3
49 !057
150 RV=25 :: CV=20 !156
160 GOTO 170 :: A$,TIME :: !
@P- !132
170 FOR TIME=1 TO 2 !031
180 CALL BAR(-RV,0,-RV,0,E,E
,C,LC,LE,LG)!059
190 CALL BAR(0,CV,0,-CV,E,E,
C,LC,LE,LG)!091
200 CALL BAR(RV,0,RV,0,E,E,C
,LC,LE,LG)!183
210 CALL BAR(0,-CV,0,CV,D,D,
D,LB,LF,LG)!090
220 CALL BAR(0,CV,0,-CV,D,E,
F,LB,LF,LG)!093
230 CALL BAR(-RV,0,-RV,0,D,D
,E,LB,LF,LG)!059
240 CALL BAR(0,-CV,0,CV,C,C,
C,LC,LE,LG)!087
300 ON TIME GOTO 310,350 !04
3
310 CALL BAR(RV,0,RV,0,D,D,D
,LB,LF,LG):: GOTO 400 !024
350 CALL BAR(RV,0,RV,0,C,C,C
,LC,LE,LG)!179
390 CALL PATTERN(#1,132,#2,1
28):: CALL MOTION(#1,0,0,#2,
0,0)!072
400 NEXT TIME !197
410 DISPLAY AT(24,11):"AGAIN
? Y" :: ACCEPT AT(24,18)SIZE
(-1)VALIDATE("YN"):A$ !076
420 IF A$="N" THEN END ELSE
DISPLAY AT(24,11):"" :: GOTO
 170 !126
9999 !@P+ !062
10000 SUB START(A$,R1,C1,R2,
C2):: CALL SCREEN(2):: FOR X
=4 TO 8 :: CALL COLOR(X,16,2
):: NEXT X !164
10010 X=(28-LEN(A$))/2+.5 ::
 DISPLAY AT(1,X):A$ !109
10020 CALL CHAR(128,"070F0A0
B0B0307070F0F1F1F3F3F3F021C1
C08FC1CFC1C1C9C88C8C8E8E8E81
8")!158
10030 CALL CHAR(132,"3838103
F383F383839111313171718E0F
050D0D0C0E0E0F0F0F8F8FCFCFC4
0")!044
10040 CALL CHAR(136,"0303010
30303030307070F0F1F1F1F02808
0008080808080C0C0E0E0F0F0F08
0")!124
10050 CALL MAGNIFY(4):: CALL
 SPRITE(#1,128,8,106,33,#2,1
32,10,106,209):: SUBEND !195
10060 SUB BAR(R1,C1,R2,C2,T1
,T2,T3,B1,B2,B3)!215
10070 P2=128-4*(P1=128):: P1
=128-4*(P2=128)!246
10080 CALL PATTERN(#1,P1,#2,
P2):: CALL MOTION(#1,R1,C1,#
2,R2,C2):: CALL SOUND(250,T1
,5,B1,10)!193
10090 CALL SOUND(350,T2,5,B2
,10,B3,10):: CALL PATTERN(#1
,136,#2,136):: CALL SOUND(25
0,T3,5,B2,10,B3,10):: P1=P2
:: SUBEND !170
```

FINDING ROOTS

The following program will find the root of a given number.

And, if you want a quicker program, here is a tinygram:

100 INPUT "NUMBER: ":N :: INPUT "ROOT: ":R :: PRINT "ANSWER:";N %(1/R) :: PRINT :: GOTO 100

ROOTS

```
100 ! ** ROOTS ** L.Dorais/O
ttawa UG/July 1989 !071
110 !!131
120 ON WARNING NEXT :: CALL
CLEAR :: PR$="PIO" !226
130 L$=RPT$("}",28):: E$=RPT
$(" ",168):: S$=RPT$(" ",8)!
069
140 CALL CHAR(120,"000000000
002050F",121,"1F102020404080
80",122,"018182C2C4646830",1
23,"0B0101",125,"FF")!167
150 DISPLAY AT(5,9):"x3y"&RP
T$("}",10):S$&"{z" :: GOSUB
280 !011
160 ACCEPT AT(6,12)VALIDATE(
NUMERIC)BEEP:N :: IF R>2 THE
N 180 !100
170 IF R=1 THEN AV=N :: GOTO
 220 ELSE AN=SQR(N):: GOTO 2
20 !151
180 AN=N^(1/R)!221
220 DISPLAY AT(12,8)BEEP:"="
;AN !203
230 DISPLAY AT(22,1):L$:" [A
]nother   [C]hange root   [P
]rint     [Q]uit" !181
240 CALL KEY(0,K,S):: IF S=0
 THEN 240 ELSE K=POS("ACPQ",
CHR$(K),1)!226
250 IF K=0 THEN 240 ELSE ON
K GOTO 260,260,270,290 !085
260 DISPLAY AT(7,12):E$:E$:E
$ :: IF K=2 THEN GOSUB 280 :
: GOTO 160 ELSE 160 !183270
OPEN #1:PR$ :: PRINT #1:S$&"
   ___________":S$&" "&STR$(R
)&"/" :: PRINT #1:S$&"\/ ";N
;TAB(26);"=";AN:"" :: CLOSE
#1 :: GOTO 240 !210
280 ACCEPT AT(5,10)VALIDATE(
"123456789")SIZE(-1)BEEP:R :
: RETURN !008
290 END !139
```

User groups sought

John Van Weelie, president of the 9T9 Toronto User Group, is seeking information for a list of active user groups he is compiling.

"I believe that good communication for the TI community will be to maintain an up-to-date user group listing," he wrote in a message posted on the Internet.

Van Weelie has a list of some 50 user groups, with current addresses. "I know that there are a lot more active user groups out there." He is looking for the following information: group name, address, city, state or province, ZIP or postal code, country, contact person, and phone number of contact.

"Once the list gets compiled I will be releasing it to TI Echo, Usenet (comp.sys.ti) on the Internet, BBSs, GEnie, and any other means."

Van Weelie asks that user groups help out by mentioning his project to groups that they exchange with, or forward a list of these groups to him.

For more information, contact Van Weelie at jvweelie@mgl.ca; or mail him at 58 Lowrey Ave. N., Cambridge, Ontario, N1R-5A3, Canada.

Hardware project

'Stealing' power for external disk drives

By JOE ZIEMBA

The following article originally appeared in the newsletter of the Pioneer Valley 99ers. We found it in MUNCH, the newsletter of the Massachusetts Users of the Ninety-nine and Computer Hobbyists. —Ed.

This modification is not recommended by most people I know. However, mine has been in place for about 5 years with no problems. I have been told that the older Peripheral Expansion Boxes (the ones with the push button on the front instead of the rocker switch used on later models) were built with somewhat better transformers and would be more likely to support the extra drive without damage. The drives I am using are the full-height models TI originally put in the PEB. Mine is a Shugart 400H. Many half-height drives draw less power than full-height drives, so they may be an even better choice for this project.

Remember, the reader is responsible for any all problems that result from attempting this project.

PARTS LIST

A full-height drive

Two 34-position insulation displacement connectors

34-conductor ribbon cable (1.5 feet long)

8 feet of 18 guage wire

Soldering iron and solder

Wire strippers

Screwdrivers

Duct tape or double-faced tape

Optional: connector that mates with the one on the second drive. That way the drive can be disconnected if a problem develops.

The thing to do is to remove the old drive from the expansion box. Then splice a wire into each of the four wires just behind the power plug (+5, +12, -12, gnd). If you are splicing on another powerplug for the second drive (this is the preferred approach), use that instead of four separate wires. Be sure that both plugs are wired the same way, or you're going to be in big trouble when you turn the system on.

If you don't want to use another connector, you can remove the power connector from the second drive and solder the power connections to the holes left on the board where the power connector was removed. Again, make sure the correct wire goes in the correct hole, or smoke city!

The logic cable is made by snapping the 34-position insulation displacement connectors onto the 34-wire ribbon cable. It is easy to do with either the tool that was designed to crimp the connectors or, as I did, by crimping them in a small vice. Be careful not to crush the connectors. Also be sure that the ribbon and the connector are lined up well before you crimp, or you may end up with a bunch of open or, worse, shorted connections.

Now it's time to set up the drive number and get the load resistors squared away. Find the drive select lines coming on to drive No. 2 — they are lines 10, 12, 14, and 16 on the logic connector for the drive. There should be a DIP switch or jumpers in each of these lines. Make sure that the jumper in line 12 (drive No. 2) is connected or turned on and that the others are open or off.

As specified in the TI disk controller manual, remove the resistor pack from drive No. 1. Put your No. 2 drive into its enclosure.

Another I did to help conserve power was to cut the tape from pin 16 on the logic connector. This is the motor-on signal.) I also removed the 150 ohm pullup resistor on this line. This was done to both drives. Then I jumpered the drive-select line (after the switch or jumpers) to the motor-on line, which is pin 16 of the logic connector (after the cut). This ensures that the motors will come on with the drive-select instead of both coming on at the same time and drawing twice the current from the 12 volt supply. You should know that a couple of the disk copying programs out there have a problem with this configuration, but most do not.

For another way of making the logic cable, refer to the December 1985 MICROpendium article "Using an internal drive externally."

Reassemble your first drive into the PEB, then connect the second drive and tape it to the side of the PEB to hold it in place. Connect the rest of the system and turn it on. How did you do? Any smoke? Did the title screen and BASIC come up O.K.? If not, shut the system down and double-check everything.

Let's assume that all has gone well up to this point. The next thing to do is to try to load a file from each drive. This is where you find out if you got your jumpers and load resistors in the right spots. If both drives responded properly, then you are done. Make sure everything is put back together neatly and that no wires are pinched.

Program released for color TIA pictures

Bruce Harrison has released a new public domain program that allows owners of 24-pin color or Canon Bubble Jet color printers to print TI-Artist pictures in color. The double-sided, single-density disk includes instructions and source code, Harrison says. Users can run the programs from Editor/Assembler Option 5, Funnelweb or Extended BASIC.

The program requires a color-capable printer of either the 24-pin ribbon type or of the Canon Bubble Jet type. It includes programs for both types.

This disk is available from the Lima Users Group, P.O. Box 647, Venedocia, OH 45894, or from Harrison at 5705 40th Place, Hyattsville, MD 20781, for $1, including media and shipping and handling.

PC99 Stage 3A released

Emulator upgrade includes host of new utilities

CaDD Electronics has released PC99 Stage 3A, an upgrade to PC99 Stage 3. PC99 is a TI99/4A emulator for an IBM or compatible computer. In addition to all features available in previous stages, the new release includes, according to the manufacturer:

• Mini-Screen trace: Allows the user to save events (for example, the last value of the program counter) to a "trace stack" and recall them. The contents of any address or register can also be traced.

• Mini-Screen instruction stack: A scrolling stack displays the last 100 instructions.

• Speech Synthesizer support: The TI Speech Synthesizer ROM is included PC99 will execute all speech functions, such as CALL SPGET and CALL SAY in TI Extended BASIC, or OPEN:#1:"SPEECH" using Terminal Emulator II, and return the correct values. Users can also run modules, such as Early Reading, and games, such as Lasso, that require the Speech Synthesizer; however, no audible speech is produced.

• Sprites: In both PC99A.EXE and PC99L.EXE, sprites no longer "flash."

• DSKDIR.EXE: Can display a disk sector map showing which sectors are used. The switches have been modified to allow for greater flexibility in outputs.

• CFG.EXE: This is now a 32-bit protected mode program, with minor changes to the menus and help screens. The user can specify an overlay file. This file can contain the text in the plastic overlay strips used with the 99/4A console. The text can be displayed to the right of the PC99 screen. If the user changes a joystick, the joystick calibration values are retained and may be reused instead of calibrating the joystick. Joystick calibration also requires the user to press Enter between stick movements to prevent "button fall-through." The joystick section also contains a test allowing the user to check calibration.

New utilities include:

• ART.EXE, which will display TI-Artist files in DOS and can be used to create a "slide show."

• DSKFIND.EXE, which will find a disk manager filename using a wild card selection of disk files.

• DLCONV.EXE, which allows the user to download files from a BBS with a TI FILES header and import them into a .DSK file using DSKIN.EXE.

• SPDUMP.EXE, which allows the user to dump the index of the Speech Synthesizer ROM and locate speech strings.

• SPCODE.EXE, which breaks down words and phrases from the Speech Synthesizer ROM into their linear predictive coding (LPC) elements.

The manufacturer says it has fixed bugs in DSKDIR.EXE, which sometimes showed the wrong file date when displaying a p-System catalog; in DSKOUTP.EXE, which no longer shows "garbage" at the end of an incorrectly extracted p-System file on side 2 of the disk, since the program now uses the last block character count; and in PC99A.EXE and PC99L.EXE, in which the BLWP instruction did not force an address to an even value. The error was found in Disk+Aid when doing a search. The cursor would become an = character, preventing text entry.

An Stage 3A upgrade is $7 if CaDD supplies the disk, $5 with the return of the Stage 3 disk. The upgrade plus two "new purchaser" disks, which allows installation of Stage 3A from scratch, is $10 if CaDD supplies the disks, $8 with the return of the Stage 3 disk. Users should specify 3.5-inch or 5.25-inch disks. Prices include shipping and handling to any country.

For further information, or to order, contact CaDD Electronics, 45 Centerville Dr., Salem, NH 03079-2674, or phone (603) 895-0119 or (603) 893-1450.

TI 99/4A CD-ROM works with PC99

Competition Computers' TI99/4A CD-ROM can now be run with CaDD Electronics' PC99 Stage 3A, according to Kyle Crichton of Competition.

He notes that all TI files on Delphi are on the CD-ROM, so these files will be available to users "no matter what happens on Delphi" regarding 99/4A access.

He says single issues of the CD-ROM are now available for $35, with a five-issue subscription $100.

For further information, contact Competition Computer, 350 Marcella Way, Millbrae, CA 94030.

Fair offers discounts, tours

Discounted advance tickets are available to the 11th Internationales TI/Geneve-Treffen Sept. 20-22 in Wolfsburg, Germany, according to Martin Zeddies, one of the organizers. Tickets are available in advance for 4 DM (5 DM at the door). Foreign visitors can pay in advance using two International Reply Coupons as the equivalent. International Reply Coupons are available for purchase at post offices.

Sightseeing tours of the Wolfsburg Volkswagen factory and of the site of the former East-West German border are planned in conjunction with the meeting, Zeddies says. He says prices for hotels in Wolfsburg range between 40 and 200 DM for one night in a single room. Persons planning to attend can have fair organizers arrange for their reservations; they will be notified of the deposit amount.

Persons wishing to purchase tickets, arrange for accommodations or exhibits may contact one of the two contact persons listed below. To receive detailed information about the fair, send two International Reply Coupons for return postage.

Contact Martin Zeddies, Haupstr. 26, D-38446 Wolfsburg, Germany, Tel/Fax: 05363 / 71125; or Burghard Knedel, Bresauler Str. 2, D-38550 Isenbüttel, Germany, Tel: 05374 / 2237.

Internet sites for the TI
FTP and home pages

The following list of Internet FTP (file transfer protocol) and World Wide Web sites for the TI99/4A was compiled by John Van Weelie, president of the 9t9 Toronto User Group. It was current as of March 30.

Van Weelie collected these addresses from Usenet, MICROpendium and other sources. His email address is by047@torfree.net. Readers who have corrections or additions may email them to him.

WEB SITES FOR THE TI-99/4A

Rich Polivka's Web Page (Ohio, USA)
http://w3.gwis.com/~polivka/994apg.html
Henry Kolpein 's Home Page (Germany)
http://sys00.ti6.tu-harburg.de/ti6hk/hobby/ti/index.html
Edward Swartz's v9t9 Emulator (Texas, USA)
http://www.io.com/~ftp/usr/edswartz/v9T9
Kerry Lee High's Web Page
http://www.umr.edu/~khigh01/994a.html
Notung Software - Ken Gilliland
http://ourworld.compuserve.com/homepages/notung/
Chris Bobbitt's Home Page - Former Asgard Software Owner
http://www.cais.com/fmg/bobbitt/clb.htm
TI FAQ Page (Texas, USA)
http://www.io.com/~vga2000/faqs/ti.faq
TI Fest West 96 - Web Page (Tom Wills - Arizona, USA)
http://www.theriver.com/TheRiver/Cafe/Calendar/fest.html
Richard Link's Home Page
http://www.iglou.com/members/ral.html
Don Shorock's Home Page
http://homepage.midusa.net/~shorock
http://www.umn.edu/nlhome/m235/walt0101/index.html

Section 2: FTP Sites Supporting TI-99/4A File Libraries and **ASSOCIATED MATERIALS**

T.M. Pederson FTP Site
ftp://solutions.solon.com /pub/ti99
Clark Net Services
ftp://ftp.clark.net /pub/ti99
Gary Bowser / OPA-NIT FTP Site
ftp://ftp.io.org /pub/users/opanit
Bruce Tomlin's 99/4A Cartridge List
ftp://ftp.crl.com /users/bt/btomlin - carts.ti99

OTHER PAGES WITH REFERENCES TO TI-99/4A OR TMS-9900.

Computer History & Emulation - Under Old Computer Section
http://www.freeflight.com.fms/comp/
Old Computer List (Mark Robinson)
http://mercury.ee.man.ac.uk/mjr/computers.htm
Chronology of Events in the History of Microcomputers
http://www.islandnet.com/~kpoulson/comphist.htm
Comprehensive Computer Catalogue
http://plato.digiweb.com/hansp/ccc/
Great Microprocessors of the Past and Present
http://www.cs.uregina.ca/~bayko/cpu.html
Obsolete Computer Museum (Tom Carlson)
http://www.ncsc.dni.us/fun/user/tcc/cmuseum/cmuseum.htm

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 OBN, 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.

MAY

Multi Users Group Conference, May 25, Ohio National Guard Armory, Brookpark. Contact Glenn Bernasek, 13246 Harper Rd., Strongsville, OH 44136, or call (after 9 p.m. Eastern time) at (216) 846-0865 or Internet dd314@cleveland.freenet.edu.

SEPTEMBER

11th International TI99/4A and Geneve Computer-Treffen, Sept. 20-22, Freizeitheim Vorsfelde, Am Sportplatz 5, D-38448 Wolfsburg, Germany. Contact Martin Zeddies, Haupstr. 28, D-38448 Wolfsburg-Reislingen, Germany. Phone/fax number +Germany-5363-71125.

1997 TI FAIRS

FEBRUARY

Fest West '97, Feb.15, San Jose, California. Contact Fest West '97 c/o Don O'Neil, 3297 Woody Lane, San Jose, CA 95132, or call (408)-934-0352.

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.

Downloading may not be easy, but it doesn't have to be intimidating

By JACQUES GROSLOUIS,

Although many printers allow the downloading of user-designed fonts, some printer owners are reluctant to try these procedures because they can be intimidating. I hope this article and the accompanying program will encourage more owners to try.

This article uses codes for the Epson FX-80 printer and may not work with other printers, even those that use codes which are described as Epson-compatible. Differences known to the author are identified. In fact, identification of more of these differences is one of the reasons for writing this article. The printer codes used by the FX-80 printer in this program are shown in Fig. 1.

This article uses codes for the Epson FX-80 printer and may not work with other printers, even those that use codes which are described as Epson-compatible.

All of the codes in Fig. 1 are used in the program which follows. In order to allow easy editing, if your printer code differs, these codes are placed in lines 110 to 190. ESC$ has been made equal to CHR$(27) and NUL$ is used for CHR$(0). The ending CHR$(0) code on the (ESC)% code is not used by the following Epson printers: FX-86e, LX-800 and LQ-850. Codes for the LQ-850 printer differ for each of the above codes. Other codes for the above are the same for the FX-86e and LX-800 printers, except that the (ESC)& code does not contain an attribute byte. Some printers allow character definition to be made in Near Letter Quality mode. This is not supported on the FX-80 printer although an emphasized mode can be selected when printing text.

Line 200 clears and sets the screen to blue and then confirms that a printer is attached. Line 210 places a title and then line 220 opens the printer port. Notice that CRLF has been added to PIO. This ensures that CR and LF codes are suppressed when printed. In Display/Variable 80 mode a CR and a LF are sent to the printer after 80 characters are processed. This would mess up character definition if these codes were not suppressed.

Line 220 continues by printing ROM2RAM$, USERAM$ and PRT128$. These codes carry out the functions for 1, 2, and 6 described above. If ROM was not copied to RAM, any characters which were not defined would appear as blanks when the RAM character set was selected for printing. Characters 128 to 159 contain the international character sets and must be activated for printing by using (ESC)6. Twelve of these characters will be redefined by this program.

The information for the redefinition of characters are entered as DATA in this program, starting at line 460. (Remember to remove the Checksums from the DATA statements before running the program.) Line 230 reads the first two items which identify the range of characters being defined. Lines 240 to 330 convert the data information to printer codes and sends them to the printer. This changes the character definitions which are in RAM. Lines 290 to 330 prints all the characters contained in RAM from 32 to 255. The redefined characters appear in line 4 of this printout. Lines 340 to 380 print a vertical bar graph.

Most printers require that they be set up to receive downloaded characters. The FX-80 printer requires that DIP switch 1-5 be OFF in order to use (ESC)&. In this position the buffer is used to store redefined characters instead of acting as a printer buffer while printing. Your printer manual should describe the required procedure to enable downloading of characters.

Another important consideration with downloaded characters is that your docu-

(See Page 25)

Fig. 1

Escape sequence	Result
1- (ESC)":"CHR$(n1)CHR$(n2)CHR$(n3) All numbers must be 0.	Copies ROM characters to the user RAM area.
2- (ESC)"%"CHR$(n1)CHR$(n2)	Select a character set: n1 selects ROM(0) or RAM(1); n2 is 0
3- (ESC)"&"CHR$(n)CHR$(c1)CHR$(c2)CHR$(A)CHR$(d1)...CHR$(d11) Each character requires an attribute byte (A), followed by 11 data numbers (d1 to d11).	Defines characters c1 to c2 in RAM area; n is 0.
4- (ESC)"6"	Enables printing of characters stored in location 128-159.
5- (ESC)"7"	Cancels use of (ESC)"6".
6- (ESC)"E"	Turns emphasized mode on.
7- (ESC)"3"CHR$(n)	Sets line spacing to n/216 inch.
8- (ESC)"2"	Sets line spacing to 1/6 inch (default).

DOWNLOADING FONTS —

(Continued from Page 24)

ment should not contain the RESET code, which is usually (ESC)@. This code would reset your printer to factory settings and wipe out your new characters.

If this program does not produce proper characters when printed, you should consider changing the codes contained in lines 110 to 190 and/or deleting the PRINT #1:NUL$ statement in line 250. This would remove the attribute bit which is used in the (ESC)& code for the FX-80 printer.

If you change "PIO.CRLF" in line 220 and place a "CLOSE #1 :: STOP" after "NEXT X" in line 330, you will produce a D/V80 file which, when printed, will load the changed characters to your printer. This is very fast and is useful if you are changing the full character set. Remember to add ".CRLF" to your printer name when printing.

The ability to download characters to your printer can allow you to change printer fonts while a document is being printed. If you combine this with the ability to include instances in your document (see article February 1996 MICROpendium) FWB editors can be used to print quite varied documents. In addition to using the ALL-CHAR characters a number of different characters can be created including converting many of the fonts developed by Jim Peterson.

If you were able to print proper downloaded characters using this program, you may be interested in a companion disk for use with Funnelweb editors which I have produced. This disk contains programs to create your own screen fonts, to convert these to printer fonts and programs to convert TI-Artist instances and output from Business Graph 99 to files which can be printed from FWB editors. The document files on this companion disk were produced from fonts and graphics produced by these programs on my FX-80 printer. For the reasons described above these programs and documents may not print properly on your printer, if some of the download codes used by your printer are different.

All the programs which download printer code are written in Extended BASIC, and changes can easily be made. I can supply this disk as a SSSD flippy or a DSSD disk, the documents are archived to save space. Since one of the main purposes of this article was to identify differences in download codes used by various printers please let me know the type of printer you use and changes (if any) you made to run this program. A few dollars to cover the cost of postage and disk would be appreciated.

Groslouis can be reached at 1747 Riverbank Drive; Bathurst, New Brunswick; E2A 4L1 Canada.

DOWNTEST

```
100 ! SAVE DSK1.DOWNTEST !17
1
110 ESC$=CHR$(27):: NUL$=CHR
$(0):: ROM2RAM$=ESC$&":"&NUL
$&NUL$&NUL$ !064
120 USERAM$=ESC$&"8"&CHR$(1)
&NUL$ !019
130 DEFCHA$=ESC$&"&"&NUL$ !2
37
140 EMPHA$=ESC$&"E" !017
150 PRT128$=ESC$&"6" :: CANE
SC6$=ESC$&"7" !037
160 !!131
170 !!131
180 !!131
190 !!131
200 CALL CLEAR :: CALL BLUE
:: DISPLAY AT(10,6):"Turn On
 Printer!" :: OPEN #1:"PIO.C
R" :: PRINT #1:NUL$ :: CLOSE
 #1 :: CALL CLEAR !007
210 DISPLAY AT(4,7):"Font Do
wn Loader": :"    By Jacques
 Groslouis" !088
220 OPEN #1:"PIO.CRLF" :: PR
INT #1:ROM2RAM$;USERAM$;PRT1
28$ !083
230 READ A,B !205
240 PRINT #1:DEFCHA$;CHR$(A)
;CHR$(B);!044
250 FOR Y=A TO B :: PRINT #1
:NUL$;!104
260 FOR X=1 TO 11 :: READ C
:: PRINT #1:CHR$(C);!217
270 NEXT X :: READ C :: DISP
LAY AT(12,1):"Creating Chara
cter #";C :: NEXT Y !186
280 PRINT #1 :: CLOSE #1 !17
2
290 OPEN #1:"PIO" :: PRINT #
1:EMPHA$ !086
300 FOR X=0 TO 192 STEP 32 :
: FOR Y=32 TO 63 !204
310 PRINT #1:CHR$(Y+X);" ";:
: PRINT Y+X;!145
320 NEXT Y :: PRINT #1:" " !
161
330 NEXT X :: PRINT #1:ESC$;
"3";CHR$(24)!141
340 PRINT #1:TAB(20);CHR$(13
2);CHR$(133);CHR$(134)!232
350 PRINT #1:TAB(20);CHR$(13
5);CHR$(136);CHR$(137)!241
360 PRINT #1:TAB(20);CHR$(13
5);CHR$(136);CHR$(137)!241
370 PRINT #1:TAB(20);CHR$(13
5);CHR$(136);CHR$(137)!241
380 PRINT #1:TAB(20);CHR$(13
8);CHR$(139);CHR$(140)!241
390 PRINT #1:CANESC6$;ESC$;"
2" :: CLOSE #1 :: STOP !227
460 DATA 129,140 !013
470 DATA 31,0,16,0,31,0,16,0
,31,0,16,129 !165
480 DATA 31,0,16,0,31,0,48,0
,64,0,128,130 !221
490 DATA 255,0,0,0,255,0,0,0
,0,0,0,131 !051
500 DATA 0,0,0,0,31,0,48,0,9
5,0,144,132 !116
510 DATA 159,0,144,0,159,0,1
44,0,159,0,144,133 !237
520 DATA 159,0,144,0,159,0,1
60,0,192,0,255,134 !236
530 DATA 0,0,0,0,255,0,0,0,2
55,0,0,135 !055
540 DATA 255,0,0,0,255,0,0,0
,255,0,0,136 !166
550 DATA 255,0,0,0,255,0,0,0
,0,0,255,137 !167
560 DATA 0,0,0,0,240,0,16,0,
240,0,16,138 !158
570 DATA 240,0,16,0,240,0,16
,0,240,0,16,139 !063
580 DATA 240,0,16,0,240,0,32
,0,64,0,128,140 !061
29505 SUB BLUE !149
29510 ! SWITCHES DISPLAY TO
WHITE ON BLUE; JLS 7/88 !230
29515 CALL SCREEN(5):: FOR L
=0 TO 14 :: CALL COLOR(L,16,
1):: NEXT L :: SUBEND !202
```

Program lets users move assembly programs or data from GRAM to RAM

By PETER DeWITTE

The program "PGRAM/O" (see Fig. 1) on the disk that I've sent may be of interest to anyone who does not have a RAM card, but has a PGRAM+ and doesn't know quite what to do with it. With a few slight modifications this program will allow you to move assembly language programs or data from any of the GRAMs on any of the pages of the PGRAM+, or page 1 of the PGRAM to expansion RAM.

For my purposes, I stored OakTree Systems' 40-column "Display Enhancement Package," DISPENFX file in page one, GRAM number 7 of the PGRAM+. For any of my programs that use 40 columns, I've included the following at the beginning of each program:

```
100 CALL INIT :: CALL LOAD("DSKx.PGRAM/O")::
CALL LINK("START")
```

This replaces the normal load procedure for the "DISPENFX" file. The advantage of course is the speed with which the file is loaded into expansion memory, almost instantaneously from Gram as opposed to from Disk.

I first loaded the DISPENFX file into memory normally and then used the PGRAM+ "hot" keys, A and Z held down simultaneously while pressing FCTN = (Quit) to enter the PGRAM without disturbing the data in low memory.

I then used the Edit Page functions of the PGRAM to move the data from addresses >2000 - >3FFF inclusive to GRAM address >E000 through >FFFF. Extended BASIC resides in the rest of Page One. Note that the CRU bits must be turned on to enable the GRAMS. This is explained on pages 13 through 22 of the PGRAM+ operating guide. The slight modifications are to the GRAM source address loaded into R2 in line number nine of the PGRAM/L list file, and modification of the GRMRD address in line number 20. The RAM Destination address in line eight, the GRAM source address in line nine and the counter (>4000) in line 26 can be changed to suit the location and length of the assembly program or data.

On a slightly different tack, in order to return to the menu screen of John Johnson's BOOT program from an *X*BASIC program automatically upon a branch to the END statement, I replace the END statement with: CALL LOAD(-31804,0,36). This tip comes from *The Smart Programmer*, Feb 84 issue.

PGRAM/S

```
0001            DEF   START
0002            AORG  >A000
0003  SAVGRM     DATA  >0000
0004  START      MOVB  @>9802,@SAVGRM  SAVE GRMRA
0005            NOP                   SEE E/A MANUAL PG 270,
0006            MOVB  @>9802,@SAVGRM+1SECT. 16.5.2
0007            DEC   @SAVGRM
0008            LI    R1,>2000        EXPANSION RAM DESTINATION AD-
DRESS
0009            LI    R2,>E000= PGRAM+ GRAM SOURCE ADDRESS
0010  *                   >6000= PGRAM+ GRAM No. 3 START ADDRESS
0011  *                   >8000= PGRAM+ GRAM No. 4 START ADDRESS
0012  *                   >A000= PGRAM+ GRAM No. 5 START ADDRESS
0013  *                   >C000= PGRAM+ GRAM No. 6 START ADDRESS
0014  *                   >E000= PGRAM+ GRAM No. 7 START ADDRESS
0015  *
0016            MOVB  R2,@>9C02       WRITE MSB OF R2 TO GRMWA REG-
ISTER
0017            SWPB  R2
0018            MOVB  R2,@>9C02       WRITE LSB OF R2 TO GRMWA REG-
ISTER
0019            SWPB  R2
0020  RPT       MOVB  @>9800,*R1+     MOVES GRMRD DATA TO ADDRESS
IN R1
0021  *                  >9800= GRMRD FOR PGRAM+ PAGE 1
0022  *                  >9804= GRMRD FOR PGRAM+ PAGE 2
0023  *                  >9808= GRMRD FOR PGRAM+ PAGE 3
0024  *                  >980C= GRMRD FOR PGRAM+ PAGE 4
0025  *
0026            CI    R1,>4000        COMPARE R1 CONTENTS TO >4000
0027            JNE   RPT             GOTO RPT IF R1 LESS THAN
>4000
0028            MOV   @SAVGRM,R2
0029            MOVB  R2,@>9C02       RESTORE GRMRA
0030            SWPB  R2
0031            MOVB  R2,@>9C02       IF YOU DON'T RESTORE THE GRM-
RA YOU
0032            SWPB  R2              WILL GET A SYNTAX ERROR ON
RETURN
0033  *                               TO BASIC OR EXTENDED BASIC
0034            CLR   R0
0035            MOVB  R0,@>837C       CLEAR STATUS BYTE
0036            RT                    RETURN TO CALLING PROGRAM
0037            END
```

Fig. 1

```
00000         9A000B0000BD820B9802BA000B1000BD820B9802BA001B06207F32FF   0001
9A012BA000B0201B2000B0202BE000BD802B8C02B06C2BD802B8C02B06C27F2C9F   0002
9A028BDC60B9800B0281B4000B16FBBC0A0BA000BD802B8C02B06C2BD8027F279F   0003
9A03EB8C02B06C2B04C0BD800B837CB045B7F7EFF                            0004
6A002START 7FD12F                                                    0005
        99/4 AS                                                      0006
```

The easy way to use the Euro-Writer mode in Funnelweb

By WOODY WILSON

Wilson is a member of the Southern California Computer Group, in whose newsletter this article first appeared.—Ed.

Have you ever wanted to use the Euro-Writer mode provided in Funnelweb V5.0, but hated to learn all the new commands? The method I'm going to outline may be what you need.

For several days I have been helping a member of the SCCG configure his Horizon RAMdisk with the 80-column version of FWB 5.0. (The instructions set forth here work for the 80- as well as the 40-column versions.)

Take a close look at the portions of the two files listed below. One is from the source code for the English text command file F4TXAE/S and the other is from the source code for the German text command F4TXCE/S.

F4TXA$/S FILE

TEXT	'SF'	Save File	
TEXT	'LF'	Load File	
TEXT	'PF'	Print File	
TEXT	'LD'	Load Temp	
TEXT	'DP'	Dir Printer device	
TEXT	'SD'	Show Directory	
TEXT	'P '	Purge	<P>urge
TEXT	'E '	Edit	
TEXT	'F '	Files	
TEXT	'L '	Lines	
TEXT	'SH'	SearcH	
TEXT	'T '	Tabs	<T>ab
TEXT	'Q '	Quit	
TEXT	'QQ'	Quit Immediate	
TEXT	'FS'	Find String	
TEXT	'RS'	Replace String	
TEXT	'M '	Move	
TEXT	'C '	Copy	
TEXT	'S '	Show	
TEXT	'D '	Delete	
TEXT	'MK'	MarK	
TEXT	'H '	Help	
TEXT	'WC'	Wildcard	

F4TXCE/S FILE

TEXT	'DS'	Save File	
TEXT	'DE	Load File	
TEXT	'DD'	Print File	
TEXT	'DZ'	Load Temp	
TEXT	'ID'	Dir Printer device	
TEXT	'I	Show Directory	
TEXT	'V '	Purge	<P>urge
TEXT	'E '	Edit	
TEXT	'D '	Files	
TEXT	'Z '	Lines	
TEXT	'S '	SearcH	
TEXT	'T '	Tabs	<T>ab
TEXT	'EN'	Quit	
TEXT	'EE'	Quit Immediate	
TEXT	'TS'	Find String	
TEXT	'TE'	Replace String	
TEXT	'ZB'	Move	
TEXT	'ZK'	Copy	
TEXT	'ZA'	Show	
TEXT	'ZL'	Delete	
TEXT	'MK'	MarK	
TEXT	'H '	Help	
TEXT	'WC'	Wildcard	

> The nice thing about this method is that, although you select German from the menu, the command lines in the Text Editor are in English.

Did you notice that the commands in the two files are arranged in the identical sequence? If you checked any of the other command text files you would find that they, too, have the identical command sequence. What does this mean to us? It means that we do not have to use the German text file to command the Text Editor when writing a letter to our German friends. We can replace the F4TXCE file by putting a copy of F4TXAE (renamed F4TXCE) on our disk in place of F4TXCE. The easy way to do this is to use DSKU and make the name change at the same time that you do the copying. If you do not have DSKU, make a temporary copy of F4TXAE to a disk, change the name of the file on that disk to F4TXCE, and then copy that file (F4TXCE) to your RAMdisk or the FWB work disk.

The nice thing about this method is that, although you select German from the menu, the command lines in the Text Editor are in English. Since the font you are using is still the German character file, all the special characters that you need are still available.

Those who write in French, Italian, Swedish, Dutch, Spanish, or other language can do the same thing by replacing that language's text command file with the English version. Here are the file names:

F4TXAE English
F4TXBE French
F4TXCE German
F4TXDE Italian
F4TXEE Swedish
F4TXEF Dutch

The Spanish text file was not on my disk, but you can make your own if you need it.

Now that you have written your letter, how are you going to print it? If you have a printer with the international character set, you can do it very easily. Most of these printers will accept a software command and will print the proper characters. If you preface your letter with the following command, you most likely will find your printer will accept German characters: CTRL-U, FCTN-R, CTRL-U, Shift-R, CTRL-U, Shift-B, CTRL-U.

Those with Panasonic printers should use Shift-A instead of Shift-B for French, Shift-E for Swedish, and Shift-F for Italian. My manual does not show a command for Dutch.

TIBUG gets new address

The Texas Instruments Brisbane Users Group has a new mailing address. The new address is TIBUG, 18 Zammit St., Deception Bay, Queensland, Australia.

DISKFILE

Disk catalog labeling in Extended BASIC

By JIRI SVOBODA

Diskfile is an Extended BASIC program that prints a diskette catalog label, which supplements a 3-ring binder storage organization.

The cataloging date is added into the standard disk manager header. Up to two labels per single page can be made by reversing a single sheet of paper. The label must be cut and inserted into the diskette slot of a vinyl page holder.

The name of a printer may be changed in line 150. Its default value is PIO.

A number of lines below the header is either 26 (this value can be changed as an upper limit of the loop in line 400) or a number of files on the diskette, whichever is greater.

The cataloging date is added into the standard disk manager header. Up to two labels per single page can be made by reversing a single sheet of paper.

DISKFILE

```
100 REM DISKFILE !005
110 REM by Jiri Svoboda !049
120 REM Toronto, Ontario !21
5
130 CALL CLEAR :: CALL SCREE
N(4):: B$=RPT$(" ",6):: TT$=
B$&"DISKETTE CATALOG"&B$ ::
CALL HCHAR(2,1,42,160):: DIS
PLAY AT(3,1): :TT$: :!054
140 DISPLAY AT(9,5):"The pro
gram generates a": :"catalog
 listing of a selec-": :"ted
 diskette and prints its": :
"disk sleeve label." !086
150 OPEN #2:"PIO",OUTPUT ::
BL$=RPT$(" ",28):: B1$=RPT$(
" ",10):: C$="|"&RPT$("-",36
)&"|" :: FOR I=1 TO 5 :: REA
D T$(I):: NEXT I !208
160 DATA DIS/FIX,DIS/VAR,INT
/FIX,INT/VAR,PROGRAM !116
170 D$=B$&"Date: " :: DISPLA
Y AT(20,1):D$ :: ACCEPT AT(2
0,13)BEEP:DT$ :: IF DT$="" T
HEN D$=BL$ ELSE D$=D$&DT$&RP
T$(" ",16-LEN(DT$))!051
200 DISPLAY AT(20,1)BEEP:"DI
SK DRIVE NUMBER:" !060
210 CALL KEY(0,K,S):: IF S<1
 THEN 210 :: CALL HCHAR(20,2
2,K):: N=K-48 :: IF (N<1)+(N
>3)THEN CALL SOUND(100,175,2
):: GOTO 210 !061
220 DISPLAY AT(23,1):"Output
 to printer":B1$;"in progres
s..." !235
230 OPEN #1:"DSK"&STR$(N)&".
",INPUT ,RELATIVE,INTERNAL :
: INPUT #1:DN$,TS,TS,AS !248
240 PRINT #2: :C$ :: CALL PL
(BL$):: CALL PL(TT$):: CALL
PL(D$):: CALL PL("(DSK"&STR$
(N)&") - Diskname="&SEG$(DN$
&B1$,1,10))!078
250 CALL PL("Available="&SEG
$(STR$(AS)&B1$,1,8)&"Used="&
SEG$(STR$(TS-AS)&B1$,1,4)&"
")!104
260 CALL PL(" Filename  Size
    Type      P"):: CALL PL("-
---------- ---- ---------- -"
)!251
300 FOR L=1 TO 127 :: INPUT
#1:FN$,FT,SN,BN :: IF FN$=""
 THEN 400 !079
310 FN$=SEG$(FN$&B1$,1,11)::
 SN$="   "&STR$(SN):: SN$=SE
G$(SN$,LEN(SN$)-3,4):: IF FT
<0 THEN FT=ABS(FT):: P$="Y"
ELSE P$=" " !247
320 IF FT=5 THEN BN$="   " E
LSE BN$="__"&STR$(BN):: BN$=
SEG$(BN$,LEN(BN$)-2,3)!050
330 CALL PL(FN$&SN$&" "&T$(F
T)&BN$&" "&P$):: NEXT L !024
400 FOR I=1 TO 26-L :: CALL
PL(BL$):: NEXT I !131
500 CALL PL(BL$):: PRINT #2:
C$:CHR$(12):: CLOSE #1 :: DI
SPLAY AT(23,1)BEEP: :"ANOTHE
R CATALOG? (Y/N)" !214
510 CALL KEY(0,K,S):: IF S<1
 THEN 510 :: IF K>96 THEN K=
K-32 !132
520 IF (K<>78)*(K<>89)THEN C
ALL SOUND(100,175,2):: GOTO
510 !046
530 IF K=89 THEN DISPLAY AT
24,1):"" :: GOTO 200 !105
540 CALL CLEAR :: CLOSE #2 :
: STOP !005
800 SUB PL(X$):: PRINT #2:"|
   ";X$;"    |" :: SUBEND !
089
900 END !139
```

```
        DISKETTE CATALOG
        Date: 041496
   (DSK2) - Diskname=
   Available=745      Used=693
    Filename  Size     Type     P
   ---------- ---- ---------- -
   AUTOEXEC      2 DIS/VAR_80
   CHARA1        5 PROGRAM
   EA           34 PROGRAM     Y
   GETKEY        2 PROGRAM
   GETSTR        4 PROGRAM
   GP550        14 PROGRAM     Y
   GPL          28 PROGRAM
   GPLREADME    11 DIS/VAR_80
   GPM          34 PROGRAM
   GPN          26 PROGRAM
   GPO          26 PROGRAM
   GPP          26 PROGRAM
   SYSTEM/SYS  481 PROGRAM
```

Creating program image files

An example of how it's done

By GEOFF TROTT

The following article is reprinted from the November 1994 TIsHUG News Digest.—*Ed.*

I have been buying the games disks from the shop (for my son, of course!) since Larry Saunders has been doing such a great job of packaging them up. For those who are not familiar, Larry puts on each disk a LOAD program which starts under Extended BASIC and then loads ROOT, which is a menu program that runs like the RAMdisk operating system MENU program. This then displays a menu of all the games on the disk which can be run by pressing the corresponding number.

This works well for Extended BASIC programs and Editor/Assembler option 5 programs but not so well for BASIC or Editor/Assembler option 3 (Load and Run) programs, as these require extra steps to be taken. It should be possible (but not easy) to change the BASIC programs to Extended BASIC programs, depending on the character sets used, and it should be possible to convert the Load and Run programs to be memory image (option 5) programs. As two recent disks (G046 and G048) contained Load and Run programs, I decided to convert these to memory image or program files. I asked Larry about them and he explained that they were too hard to do in the time he had available, so this seemed to be a good thing for me to try to do.

The first thing I did was to examine these two programs. One of these was called Caverns and consisted of two files called CAVERNS1 and CAVERNS2. These files were Display/Fixed 80 types and the first one was about 48 sectors long with the second one 93 sectors long. This program did not self-start so a starting name had to be typed in.

The loading process starts with the menu number which loads a file called ED/AS. This then shows the Editor/Assembler menu with item 3 labeled CAVERNS. Pressing 3 brings up the prompt to enter the first file and the cursor has to be moved back to change the disk number from 5 to 1 and then type in the file name CAVERNS2.

When this file has been read in, there is a prompt for another file, this time from DSK1. After typing in CAVERNS1 and waiting for it to read in, pressing enter at the next prompt causes the start name of CAVERN to appear and pressing enter again gets the program running. I have not been able to get the program running by putting the files in the other order unless I use Funnelweb or the Editor/Assembler module.

The second of these programs is called Perfect Push. It is one Display/Fixed 80 file of 91 sectors long. It starts the same way as Caverns up to typing in the first file name, except the file name is PUSH, and it starts automatically without any further typing. I first decided to make Caverns autostart, which worked OK but still required the two file names to be typed in so this is part of another story.

I then decided to look at the files for these two programs and work out how to save them as program files. Saving programs as program files is done by using the SAVE program, which is a Display/Fixed 80 file itself. SAVE requires that some symbols are defined before it is run. These are SFIRST, SLAST, SLOAD.

It should be possible (but not easy) to change the BASIC programs to Extended BASIC programs, depending on the character sets used, and it should be possible to convert the Load and Run programs to be memory image (option 5) programs.

SLOAD defines the first address of the program and SLAST the last address the program uses in memory. SFIRST must be the same as SLOAD and, I assume, was originally designed to be the starting address of the program but because of the constraints of the program file format cannot be different from SLOAD. This means that the first instruction that will be executed is the first thing that is loaded into memory and so is the first data loaded into memory in the first file of the sequence of files that make up the memory image version of the program.

Memory image files can be more than one in number because they are restricted to 33 sectors (8 Kbytes) in size (normally, anyway), or because there are gaps in the memory used by the program (each file contains contiguous memory locations only). The files that will be loaded as a continuous program have names whose last character increases in its ASCII code by one, file by file.

The SAVE program does this for us in the simplest cases. Funnelweb's FSAVE does it all a little better. FSAVE has several entry points: SAVE, which does the normal save function, but a little better; MBSAVE, which adjusts the high memory start to >A050 above the Mailbox; and MEMSAV, which allows the memory address limits to be entered, irrespective of the values in SFIRST and SLAST. There is good detail about FSAVE in the Funnelweb documentation on Utilities.

Back to the cases in point. When I examined the Caverns files using Funnelweb's DiskReview, I found that they were compressed format and used absolute addressing and data. To do this, I used the Inspect file function of DiskReview, looking at the offset in the file. This allowed each sector of the file to be looked at in hexadecimal or in ASCII.

Since the file is in Display/Fixed 80 format, each record is exactly 80 bytes (characters) long, which is >50. Since there are 16

(See Page 30)

IMAGE FILES —

(Continued from Page 29)

items on each line of the display (whether in ASCII or hexadecimal format), each record takes up five lines and there are three records in each sector. Each record consists of a number of character tags followed by up to two fields. These character tags are ASCII characters, while the data in the fields are in binary. The most used character tags are 9 for absolute addresses, B for absolute Data, A for relocatable addresses, C for relocatable data and F for the end of record. The first character tag in each record is an address tag, and this is followed by a number of data tags indicating words of data (two bytes).

Looking at a sector in ASCII mode and seeing lots of 9s and Bs means that the code is all absolute. To find the range of addresses used by a Display/Fixed 80 file, go to the first sector and find the first 9. Position the cursor on this 9 and change to hexadecimal mode. The address is then the four digits after the 39 on which the cursor is resident.

To find the last address, go to the end of the file and look for the last 9 tag (in ASCII mode). This will be followed by a number of records with 3, 4, 5 and 6 tags, indicating External references and External definitions. Having found the last 9 tag, count the number of B tags following it, determine the address following the 9 tag and add twice the number of B tags following it (in hexadecimal, of course) to this address. The Editor/Assembler manual explains more about these tags.

Using this approach, I found that CAVERNS1 contained data for addresses >D400 to >ED7A while CAVERNS2 contained data for addresses>A000 to >D3BA. CAVERNS1 had entries for ROCK, HAT and MINE while CAVERNS2 had an entry for CAVERN, the starting address at >A000.

I thought that this would be quite easy to convert to program format as it satisfied the requirement of the first data also being the first instruction to be executed. However, it did have references (REFs) to routines like KSCAN, VDPWA, VDPWE, VDPRD which may require the presence of the Editor/Assembler cartridge. That is something to be kept in mind. The first thing is to produce the program format file for the program itself. What is missing is the definition of the names, SLOAD, SFIRST,SLAST.

From looking at the data with DiskReview, I know what values these must have so it is only a matter of making the loader aware of the values. To do this a small assembler language program needs to be written to define their values. This looks like the following:

```
            DEF   SFIRST,SLOAD,SLAST
SFIRST EQU  >A000
SLOAD EQU   >A000
SLAST       EQU   >ED78
            RORG
            BSS   SLAST-SFIRST
            END
```

The first line says that the names listed are defined in this program. The next three lines define the values for the three names. The next two lines change the pointers used by the loader to point to the end of where the absolute code ends. This is so that when the save program is loaded, it will no load on top of the program you are trying to save. This is because the loader does not update the pointers for absolute code. If the code were relocatable (tags of A), these two lines would not be needed and SLAST could be determined another way.

This program needs to be typed in, assembled and then loaded. I called my program PFF;S for the source and PFF;O for the object file (D/F80). The saving sequence was then:

- Choose LOADERS option from the Funnelweb menu
- Choose Load and Run from the next menu
- Type in DSK1.CAVERNS2, then DSK1.CAVERNS1, DSK1.PFF;O, DSK2.FSAVE (Funnelweb disk in DSK2), and then a blank line.
- Choose SAVE from the list of possible entry points, press CTRL-A and then type in the file name you wish to use (DSK1.CAVERN).

Three files are produced, CAVERN, CAVERO and CAVERP. This can be tested to see if it needs the Editor/Assembler cartridge by loading it using the Funnelweb loader in GPL mode. It worked fine like this so it obviously uses the functions available in Extended BASIC.

The second program was a bit trickier as it did not start at the first address (>A000) and it automatically started, which does not allow SAVE to be loaded and run. Looking at the Display/Fixed 80 file with DiskReview towards the end of the file, we need to find the record which has the tag character of 1. This is the entry point and needs to have the tag character changed to F to disable this feature.

I found that the program used absolute addresses and memory from >A000 to >DCCE and had a starting address of >ADAA. Talking to Rolf about this, he said that his normal solution would be to note the contents of the first two words at >A000 and >A002 and replace them with a B @>DCCE. At the end of the program he would put in code which would restore the contents of the memory locations changed and then jump to the starting address.

That is a clever way to do it, but I thought of a simpler way by making use of the facilities of FSAVE. My idea was this: load all the files into memory as before (change PFF;S to suit and assemble); choose MEMSAV as the starting address and enter the addresses as from >ADAA to >DCCÎ and the file name as PUSH0. this will generate two files, PUSH0 and PUSH1. After doing that, FSAVE returns to the loader and entry of a blank line followed by starting at MEMSAV again allows the rest of the program to be sent to PUSH2 from >A000 to >ADA9.

Now the third file needs to be linked to the first two by using DiskReview to Inspect the first sector of the file PUSH1 and change the first word in the sector from >0000 to >FFFF. Now the result can be tested by loading via Funnelweb. In this case the program did not work in the GPL environment but did work in the Editor/Assembler environment. So it is necessary to also load in the Editor/Assembler code into low memory. Funnelweb has a file called EA which is just what is required, but it needs some doctoring.

(See Page 31)

IMAGE FILES —

(Continued from Page 30)

First copy EA into the disk with the three other files and rename it as PUSH3. Inspect the File with DiskReview and change the second word to >07FA and the third word to >2000 being the site and address of the file. The Inspect PUSH2 and change its first word from >0000 to >FFFF to allow the fourth file to be linked with the other three. Then all works well when loaded with any environment. It is a bit difficult to describe all the details in words, but if you have a small idea of what is going on, it should be easy to do this sort of thing with any program.

As a postscript, I also noticed that on one of the games disks (G032), there is another version of TI-Runner which starts with 30 men. This is supplied as two program files (RUN, RUO) and a Display/Fixed 28 file (LEVEL28) of the different screens. However, it is also run through the ED/AS program, which made me think that it needs the Editor/Assembler cartridge environment to run. So, I copied the PUSH3 file to this disk, renamed it RUP and changed the first word in the first sector of the RUO file from >0000 to >FFFF. Then the program loaded and ran without the need for the ED/AS file.

Deaths in the TI family

Death of Shawn Baron ends Red Baron Software

Shawn Baron, age 17, died March 25 of the complications of influenza. Shawn, the son of Richard and Nancy Baron, was born Oct. 27, 1978. Shawn had been ill for about a week, when his heart failed. The doctors were able to revive him twice, but not after the third stoppage.

Shawn, along with his father, Richard, had formed the Red Baron Software, a new up and coming TI software producer. All the software sold was written by Shawn and Richard Baron, and was all new, original, software. Red Baron Software made its premier showing at Fest West '96 held in Tucson on February 17, 1996.

Richard Baron recently posted the following notice on the Cactus Patch BBS of the SW99ers:

"Due to the death of Shawn Baron, Red Baron Software is no longer in business. This was because from 75 percent to 90 percent of the programming was done by Shawn. It will take me a number of years to attempt to get anywhere near the level of understanding that my son had of programming. I'm not even sure if I will be able to program anymore. If so, it probably be a number of years before I could get anything done."

He thanked customers and said that Shawn had also been grateful.

Shawn, along with Richard, had been a long time member of the SouthWest Ninety Niners User Group of Tucson. Shawn was special to the SouthWest Ninety Niners, Tom Wills of the group notes, "He always tried to help others. If questions involved the assembly language, he almost always knew the correct answer."

Shawn's funeral was held at the St. Francis de Sale Catholic Church in Tucson March 29.

The Southwest Ninety Niners User Group will be collecting donations to help offset the expenses of Shawn's Hospital stay and Funeral. Anyone wanting to contribute can send a donation to: SouthWest Ninety Niners User Group, P.O. Box 17831, Tucson, AZ 85731-7831. The group asks contributors to note on the check that the donation is for the Shawn Baron Fund. Condolances can be sent to: Richard and Nancy Baron, 8427 East Crambria Drive, Tucson, AZ 85730

Frank Bunzel dies

On April 4, 1996, Frank Bunzel, a long-time member of the Sheboygan (Wisconsin) Area 99ers User Group died after a long illness. This is the second long-time Sheboygan member to die in the past year. Last year Joseph Turk died unexpectedly.

Frank, age 75, of 1806 South 19th Street, Sheboygan, was born Feb. 15, 1921, in Sheboygan Falls, Wisconsin, to the late Paul and Pauline Bunzel. Frank had been employed at the Kohler Company of Kohler, Wisconsin, served as a deputy sheriff for the Sheboygan County Sheriff's Department, and for the last 30 years as a car salesman for variou dealerships in Sheboygan until his retirement.

Frank is survived by his wife of almost 48 years, Elaine, and his daughter, Kristine Bunzel

Frank served with the US Marines in the South Pacific theater in World War II from 1942 until 1946. A memorial service for Frank was held at the Ramm-Ziegler-Novack-Rettke Funeral home with the Rev. Brian Szyszko officating.

A memorial fund has been established in Frank's name at the St. Peter Claver Catholic Church in Sheboygan. Frank has asked that his TI equipment be donated to schools. Leon Kreidler of the Sheboygan Area 99ers User Group is heading up the distribution of the equipment to schools that will make good use of it.

Tom Wills, who sent the notice to MICROpendium, notes, "Frank is another TIer that I will dearly miss. I have always kept in touch with Frank over the years since I left Sheboygan via letters, BBS messages, and visits whenever I was in Sheboygan. Frank Bunzel was one of those TIers that made you glad to be a TIer yourself. Frank, you will be missed, but your memory will live on and you will still have a great influence on the school children who will continue to use your beloved TI equipment."

Merideth Beyers dies

Merideth Beyers of the Southern California Computer Group died April 27, two weeks after his 97th birthday, according to the group's newsletter. He was the author of the series "Playing with Numbers" which began appearing in the newletter in November 1990. According to the newsletter, the articles in the series are being assembled into a booklet.

He is survived by his wife.

Chainlink Solitaire

A solitaire program that challenges the experts

Edited by LARRY SAUNDERS

This comes from the April 1996 issue of the TIsHUG News Digest.—*Ed.*

Chainlink Master Solitaire © 1989 Walt Howe and Wayne Stith. Design: Walt Howe. Program Implementation: Wayne Stith: Documentation: Walt Howe.

Chainlink is one of the most challenging of all solitaire card games. It offers a high solution rate if played very well, but it never ceases to present a challenge to the good player. About 50 percent of all games are solvable, but for the beginner, it may take many games before the first solution comes. When Chainlink is dealt out on screen, the entire deck of cards is visible. There are no hidden or face-down cards. With all the cards visible, there is no luck factor after the cards are dealt. It entirely depends on your skill in playing.

Chainlink is one of the most challenging of all solitaire card games. It offers a high solution rate if played very well, but it never ceases to present a challenge to the good player.

OBJECT OF THE GAME

The object, as with most solitaire games, is to build all the cards in order from ace to king in each suit. Cards are played one at a time, beginning with the aces, into four piles at the top of the screen, one for each suit. When you have all 13 cards in each suit played to the four top piles, you have won the game.

THE LAYOUT

The cards are initially dealt into 13 columns of 4 cards forming the card layout or tableau. Each column is labeled with a letter, ranging from A to M.

THE RULES

The rules are simpler than most solitaire games. These rules apply to play with actual cards as well as on the computer.

1. Only one card may be played at a time.

2. Only the bottom card of any column may be played. This applies to cards played within the tableau and to cards played to the top four piles. No cards can ever be played from the middle or top of a column until the cards below it are played first.

3. In the tableau, a card may only be played directly on another card of the same suit and one less or one greater in sequence. For example, the six of diamonds may only be played on the five of diamonds or the seven of diamonds. The six of diamonds must be the bottom card of its column, and the five or seven must be the bottom card in its column, also.

4. The top four piles must begin with the aces of the four suits. They may be played any time they are the bottom card of a column in the tableau. Subsequent additions to the top four piles must be in order from ace to king. Any time the card at the bottom of a column is the next card in sequence for the top pile of its suit, it may be played to the top pile.

5. Empty columns may only be filled by kings. Any king can fill any empty column, but only if it is the bottom card of another column.

6. No column can contain more than 10 cards.

7. A deal may be restarted from the beginning position at any time. (Of course, this rule is virtually impossible with an actual deck of cards.)

8. The last card played may be taken back to its former position at any time until another play begins.

9. These rules are enforced by the computer. It is not possible to cheat in this solitaire game.

GETTING STARTED

The program loads from the Editor/Assembler cartridge (or equivalent) or from Extended BASIC. From Extended BASIC, place the disk in drive 1 before you select Extended BASIC and it will load automatically. If you are already in Extended BASIC before inserting the disk, type RUN "DSK1.LOAD" with the Chainlink disk in drive 1.\

From the Editor/Assembler cartridge, select option 5 and type in the filename DSKn.CHAINLINK, where "n" is the number of the drive. Other options you can use are TI-Writer option 3, the Funnelweb program image file loader or similar loaders. When the program is loaded, the title screen is displayed. Simply press the Enter key to begin play. The cards are then dealt and the menu appears below the tableau. The options are as follows:

1. *Begin.* If you select this option, the menu is cleared from the screen and all possible moves to the top piles are automatically made for you.

2. *Redeal.* If you do not like the current deal, you can select another deal with this option.

3. *Read rules.* An abbreviated, but complete, set of rules and key press instructions is available by selecting this option. When you have finished, the same deal will be displayed again.

4. *Save game.* If you want to save the deal for future use, you can select this option and you will be prompted for a filename. You will have later chances to save the game, also.

5. *Load game.* You can restore a saved game with this option. When you select it, you are prompted for the name of the saved file.

6. *Quit.* This option exits the game without saving the current deal.

MECHANICS OF PLAY

All play is controlled by key presses, as follows:

(See Page 33)

CHAINLINK —

(Continued from Page 32)

1. All possible plays to the top piles are made any time you press the Enter key.

2. Plays within the tableau from one column to another are controlled by pressing first the letter of the column you want to move a card from and then the letter of the column you want to move to.

3. If you want to repeat the same move when you are moving a series of cards in order one at a time, just press the R key.

4. If you want to take back a move, press Function 1. This is known as the OOPS key, for those cases where you pressed the wrong key. Once you have pressed another key, the OOPS key will not work, whether or not you have actually moved another card.

5. Any time you want to save a game to disk, press Function 2. It will save the original deal, but not the position of the cards after play has started. You can abort a save by pressing Function 4.

6. Any time you want to reload an old deal, press Function 3. You can abort the load with Function 4.

7. Your first deal in any session is dealt visibly with moving cards. If you get tired of seeing the cards dealt out, press Function 5 while the menu is still on the screen and you will toggle off the visible deal. To restore it, press Function 5 again while the menu is on screen.

8. When you are stuck in a game or just want to quit, press Function 8. It will display your score in won games and percentage of cards played and ask if you want to save the game. Next it will ask if you want to play again. If you say Y for yes, it will deal a fresh deal for you to try.

9. Any time you want to restart the current deal, press Function 9 (the Back key). It will redeal the same deal and display the menu again.

STRATEGY

The name Chainlink comes from the strategy of building a chain of cards in sequence in the tableau. It is often more important to build sequences than it is to play cards to the top piles. A linked chain is one which has its two ends in different columns, allowing you to move either way. Try to break the two-column link only when you are working to extend the chain further. There will be many cases when you cannot do this, of course, but only break the two-column structure when you have to. Another desirable position is to have inverted sequences with a king at the bottom and one or more columns empty. This is another type of linked chain, since the king can move into the empty column at any time. Even if you have several sequences with kings at the bottom, one empty column gives you a lot of flexibility. When you have one or more kings at the bottom and no empty columns, developing an empty column should be a prime goal.

When you can see no way to extend chains, then work on freeing aces and subsequent cards to the top piles. The more cards that are played to the top, the simpler the remaining play becomes.

Never play a card on a column when both cards that it can move onto will both be in the same column beneath it. Sometimes an initial deal will include this arrangement. If you see this arrangement, redeal immediately. The game cannot be solved. A similar stuck position is when the interior cards of a sequence of four are at the bottom of columns and the exterior cards are above them in the same columns. If you recognize these positions while the menu is still on the screen and redeal, the game does not count against your total score.

Set yourself goals as you improve your play of the game. While you are still learning, just solving a game is a triumph. Later, you will want to see how high a card score or game score you can get in a series. Try for 5 wins in 10 games or 10 wins in 20 games; 13 wins in a 20-game series is outstanding! Can you top it without using the Function 9 key at any time?

THE CHALLENGE GAME

The game disk includes 50 saved games that are all guaranteed solvable. Some are very easy; some are extremely difficult. Can you solve all 50 games?

CHAINLINK
NOW PUBLIC DOMAIN!

Chainlink, formerly a commercial product of Genial Computerware, has now been placed into public domain by the authors, Walt Howe and Wayne Stith. Enjoy!

MICRO-REVIEWS

Virus Attack, Word Search, Printer's Apprentice and Toolbox Tutorial

By CHARLES GOOD

VIRUS ATTACK
by Vern Jensen

This is another really good game from an author who has a track record of good 99/4A games. Vern wrote The Castle. His newest game, Virus Attack is a near clone of the Dr. Mario game cartridge for Nintendo game machines.

The game vaguely resembles Tetris. There are falling rectangular blocks of one or two colors. When you get four square color blocks (either four two-color blocks with the same color on one end or two one-color blocks of the same color) lined up either vertically or horizontally then the four color squares disappear. At this point anything on top of the disappeared blocks

(See Page 34)

MICROREVIEWS —

(Continued from Page 33)

falls into the newly created void and points are awarded. As the rectangular blocks slowly fall they can be maneuvered left/right and rotated in various ways with the joystick. You win if you finish the 20th level or if you earn 10,000 points. There are three skill levels, with the blocks falling faster with successive levels. You select a skill level and the entire game is then played at that speed.

For an action game such as Virus Attack a simple description of the game gives the reader no real indication of whether the game is enjoyable. Perhaps you really don't understand my description in the above paragraph. Seeing is believing. If you like action games then when you see Virus Attack I think you will like it. I know my 15 and 17 year sons and their friends really like it. Both my sons have spent hours and hours playing. Their friends come over to the house specifically to play Virus Attack. It is the kids' current "most favorite TI game."

The game is written in assembly language and has very fast and responsive joystick action. You can also play from the keyboard. Graphics are interesting and colorful. If the action gets too fast you can pause the game. There is neat background music, which can be toggled on and off, and many sound effects.

Virus Attack comes on a single-sided, single-density disk with instructions readable from within the program. It boots automatically from DSK1. with Extended BASIC. The game is shareware, and the author asks $15 plus $2 shipping if you order it directly from him. I recommend that you order directly from the author because I know you and your kids will like Virus Attack. However, in case you want to check it out first and send the $15 later, I will send you the disk for $1 including shipping.

If you like action games then when you see Virus Attack I think you will like it. I know my 15 and 17 year sons and their friends really like it. Both my sons have spent hours and hours playing.

WORD SEARCH
by Bruce Harrison

This is another assembly language public domain offering from Bruce. It prints word search puzzles to your printer that look very similar to those published in magazines and books.

As you may know, word search puzzles hide a list of words in a matrix of printed letters. At first glance you appear to be looking at a block filled with random letters. The object of the game is to recognize real words hidden within the otherwise randomly printed letters in the box. Words can be printed left-to-right, right-to-left, up-to-down, down-to-up, or diagonally. These puzzles are hard! At least they are to me.

Bruce's program uses words entered in a DV80 text file, one word per line. The words are randomly arranged in a 25x25-character matrix, and extra spaces in the matrix are randomly filled with letters that are not part of the word list. A full matrix can accommodate between 50 and 70 words depending on average word length. You can use Bruce's predefined word lists, or make your own lists.

When you run Word Search you are asked for the word list file name, which can be loaded from any path including hard disks. Once loaded, the words quickly scroll down the screen for your inspection and then you are asked for your printer name. Usually you would accept the default "PIO," but you can also specify a disk file instead if you want to save your puzzle on disk. After entering a printer name the printer immediately begins printing the puzzle. Because this is an assembly language program there is no waiting for the computer to "think" about how it will arrange your words in the matrix.

The matrix is printed centered in the upper two-thirds of the page. Below the matrix the sentence "Find these xx words in the puzzle above." is printed, where xx is the number of words in the puzzle. Then the words are printed in alphabetical order in five neat columns. Below this is the sentence "Made using the file DSKx.FILENAME" specifying the path and name of the source file. All of this printing goes on one 8.5x11-inch sheet of printer paper.

Word Search comes on a DSSD disk with versions for both 9-pin and 24-pin dot matrix printers and a bunch of predefined word lists. You can also, of course, use your own word lists created with a word processor. Topics for the predefined word lists include American Revolution, birds, British cities, Canadian cities, cars, colleges, countries, dog breeds, English towns, fish, flowers, French Revolution, foods, games, German places, Italian places, Irish places, islands, lakes, mountains, music, northern Welsh towns, rivers, Scottish towns, southern Welsh towns (Welsh towns have really unusual names), sports, states, teams, trees, and bodies of water. I also have a single-sided, single-density version with fewer word lists. Send me $1 and I will mail you either the DSSD or SSSD Word Search disk.

PRINTER'S APPRENTICE
AND TOOLBOX TUTORIAL
by Marv Smith and the B.C. 99ers.

The most full-featured desktop publishing software available to 99/4A users is The Printer's Apprentice and its companion software, TPA toolbox. With these two software packages you can create new graphics or use TI-Artist pictures anywhere on the page. You can create your own text within the program or import TI-Writer text and print that text in any combination of TI-Artist or Character Sets and Graphic Design fonts with the text micro right-justified. No other 99/4A page creation software does micro right justification. Because of the 99/4A's limited memory TPA can only load parts of itself into memory, and this is done manually by the user. Since TPA is so full featured it is also

(See Page 35)

MICROREVIEWS —

(Continued from Page 34)

very complicated to use, and unfortunately the documentation that comes with TPA is very difficult to understand. The poor quality of the original TPA documentation has resulted in the publication of two third party TPA tutorials. (Maybe it is "fourth party," since TPA is "third party" software.) I have in the past reviewed Ken Gilliland's excellent and humorous TPA tutorial geared to users of both the Geneve and 99/4A versions of TPA. Now Marv Smith has written a second tutorial specifically for the 99/4A TPA user.

Marv's tutorial is very appropriately labeled "A B.C. 99er Tutorial," referring to his membership in the Vancouver British Columbia User Group. The tutorial walks you through the ABCs of all 10 separate programs that make up the 99/4A version of The Printer's Apprentice and the TPA Toolbox. TPA and TPA Toolbox are sold separately and the tutorial assumes you have both of them. You really do need both parts to do useful work. The tutorial comes with a disk of sample files, something not found with Ken Gilliland's tutorial. Following the step-by-step tutorial instructions and the sample files you will print a page of TI Base information (a TI Base tutorial within the TPA tutorial) with a picture near the center of the page surrounded by seven different combinations of fonts and type sizes. Most of the text will be microjustified the page will print in one continuous and very slow print operation.

One of the things I really line about Marv's tutorial are the sample printouts. You are shown on paper how one font can be modified 12 ways (printed darker or lighter, squeezed or stretched in various ways) by the Toolbox font converter program. You are shown each of the 20 border patterns the toolbox gives you to modify, and you are given printouts of some sample forms made with the toolbox's form maker. These printouts are all useful.

If you want to completely create newsletters, flyers or brochures without cut and paste and using only a disk-based 99/4A and printer, then TPA and TPA Toolbox make a serious and perhaps better alternative to Asgard's Page Pro. Check out the documentation that comes with any of Ken Gilliland's Notung software products to see what a "made completely with TPA" product looks like. If you want to use TPA than you will probably require supplementary documentation such as that reviewed here. Marv Smith's TPA tutorial with disk supplement is available for $10 US plus $2 shipping from the B.C. user group. The TPA software itself for the 99/4A and TPA toolbox can be purchased for $22.50 each ($45 total) from Ramcharged Computers. Two sets of supplementary TPA font disks are also available from Ramcharged.

ACCESS:

Vern Jensen (Virus Attack): 910 Linda Vista Ave., Pasadena, CA 91103.

Bruce Harrison (Word Search): 5705 40th Place, Hyattsville, MD 20781. Phone (301) 277-3467

B.C. 99er User Group c/o Ron Warfield (Marv Smith's Printer's Apprentice and Toolbox Tutorial): 216 10th Ave., New Westminster, British Columbia, Canada V3L 2B2.

Ramcharged Computers (a dealer that sells the TPA and TPA Toolbox software): 6467 E. Vancey Dr., Brook Park, OH 44142. Phone (216) 243-1244.

Charles Good: P.O. Box 647, Venedocia, OH 45894. Phone (419) 667-3131. Internet e-mail cgood@osulima1.lima.ohio-state.edu (preferred) or good.6@osu.edu.

Myarc becomes division of Cecure

Cecure Electronics has acquired Myarc Inc., the company which produced the Geneve 9640 computer and other products.

The acquisition was accomplished May 1.

Cecure has been the authorized Myarc repair center since Myarc went out of business in 1992.

Don Walden of Cecure Electronics says that Myarc is now a division of Cecure Electronics Inc. (CEI).

"We will be marketing under that name all products that were produced by Myarc and ones that were under development but never got produced," he says.

He says Cecure will continue to sell Geneves, Hard and Floppy Disk Controllers, 80-track Floppy Disk Controllers, RS232 32K, 128K 256K and 512K RAM/memory cards.

He said the company received about 600 pounds of "cards, gate arrays and blank boards," and that more material is on the way. "We have more than 1,000 gate arrays," he says.

Walden says the new materials received should make quicker turnaround time on repairs to Myarc products possible, as simple swaps for working components will now be easier to do.

He says that Tim Tesch is working on enhancements to MDOS, and also working with putting the Rave memory card in the Geneve system as a RAMdisk.

The Geneve, a 99/4A-compatible computer, has an 80-column screen and greater speed and memory than a TI99/4A. It was developed by Lou Phillips, founder of Myarc.

Fest West set for San Jose

Fest West '97, sponsored by the South Bay TI Users Group is scheduled for Saturday, Feb. 15, 1997, at a downtown hotel in San Jose, California.

Admission for adults is $4, with children under 16 free. Numerous hotels in the area are available, according to Don O'Neil, one of the organizers.

O'Neil say the Fest will be "in the heart of Silicon Valley" convenient to restaurants, theaters, the Tech Museum of Innovation and the Childrens Discovery Museum.

For information and a Fest package, call (408)-934-0352 or write: Fest West '97 c/o Don O'Neil, 3297 Woody Lane, San Jose, CA 95132.

USER NOTES

The TI99/4A typewriter

The following article has appeared in several newsletters. It was written by Charles Ball.

There are often times when we just want to type a short note or letter and, rather than load in a full-blown word processing program, we settle for writing with such low-tech implements as pens and pencils.

It is easy to turn your printer into an electric typewriter. Four lines of BASIC code will do it.

This program enables the user to type a line of text, edit it as desired, and then print it by hitting the Enter key.

Whenever a line of text is to be indented or contains a comma, that line must begin and end with quotation marks. The quotes will not be printed, nor will they be counted in the length of the line of text.

The skip a line, just hit Enter.

The short program No. 1 allows sending of printer codes directly to an Epson RX-80 printer.

By adding a few more lines, the program can be made even more useful. We can require an input as to the maximum width that is to be printed and use this information to set equal right and left margins. A check has been added to ensure that the maximum line length is not exceeded, and it includes a prompt to display what an overly long line can be shortened to. User instructions have also been added. The expanded 10-line BASIC program is listed as program No. 2.

When typing notes, etc., where it is desirable to start printing at column one, input a line length of 80 and monitor the screen for your line length.

A simple way to use this program for correspondence is to use a line length of 56. This will fill exactly two lines of the TI screen. Right margin justification can be accomplished by inserting spaces between words until the second line of text is completely filled.

Of course, the OPEN statement in line four should be changed as required for the printer you are using. The line length feature is designed for pica print. Line three can be changed to accommodate elite or condensed type styles.

PROGRAM NO.1

```
1 OPEN #1:"PIO"
2 INPUT A$
3 PRINT #1:A$
4 GOTO 2
PROGRAM NO. 2
1 PRINT : : : :"TO INDENT TE
XT OR USE A     COMMA, BEGIN
 & END THAT LINEWITH QUOTATI
ON MARKS": :
2 INPUT "PRESS ENTER TO SKIP
 A LINE. HOW WIDE?(80 CHARAC
TERS MAX)":WIDTH
3 MARGIN=INT((80-WIDTH)/2)
4 OPEN #1:"PIO"
5 INPUT "INPUT A LINE OF TEX
T:       ":TEXT$
6 IF LEN(TEXT$)>WIDTH THEN 7
 ELSE 9
7 PRINT :"LINE TOO LONG! SHO
RTEN TO": :WIDTH;"CHARACTERS
 MAX.": :SEG$(TEXT$,1,WIDTH)
8 GOTO 5
9 PRINT #1:TAB(MARGIN);TEXT$
10 GOTO 5
```

Adding characters in condensed mode

Bruce Harrison of Hyattsville, Maryland, wrote this in response to a Reader-to-Reader plea from Martin Zeddies of Wolfsburg, Germany, for some simple way to make TI-Writer (or Funnelweb Text Edit) allow more than 80 characters per line on the printer's condensed mode. Harrison writes:

The following method will work with most dot-matrix printers, but not all, so a simple test is recommended.

First, through whatever method is needed, put in the control sequence to place the printer in condensed mode. This will allow 136 characters per line on the printer, but the trick is to avoid the carriage return and line feed after each 80 characters when printing.

Each line of 136 characters on the paper will occupy 1.7 lines in the TI-Writer document. Thus, you'd alternate between lines of 80 and lines of 56 characters in length.

Type to near the end of the first line, then hit CTRL-U. Now hit Shift-S, then Enter to get to the next line, then Shift-Q, then CTRL-U to get back into normal typing. On this even-numbered line type only 56 or fewer characters, then a carriage return. The Shift-S while in CTRL-U mode puts a CHR$(19) (Hex 13) in the document, while the Shift-Q puts a CHR$(17) (Hex 11) at the start of the next line. On most dot matrix printers, that Hex 13 will put the printer off line, so it won't recognize anything except the Hex 11 control character. Thus, the printer will ignore the carriage return and line feed sent by the formatter, and will continue printing with whatever comes after the Hex 11 on the same printing line.

Just continue in this manner with alternating long and short lines for as long as you want to stay in condensed mode with lots of characters per line. This method will work on most models of Star Micronics, Epson and Panasonic printers. It won't work on some older models, most notable the Epson RX-80, which does not recognize the off-line and on-line control characters. On some printers, you can use a combination of elite and condensed modes to get 160 characters on each printed line, using the same trick to avoid the unwanted CR/LF actions.

Using CHARA1FIX with FWB characters

This comes from Jacques Groslouis of Bathurst, New Brunswick, Canada:

CHARA1FIX was written by Wayne Stith and appeared in the June, July and August 1989 issues of MICROpendium. If you attempt to use this program with screen fonts included with the Funnelweb Editors a very annoying problem arises. Although these fonts if renamed as CHARA1 will load, the problem arises when you attempt to save changes back to the file you have loaded. The program seems to be with the six-byte header which is at the start of a CHARA1-type program file. A CHARA1-type file from FWB contains a header which is usually 0000 0000 000. In order to use

(See Page 37)

USER NOTES

(Continued from Page 36)

CHARA1FIX, the first two bytes is a flag word and is usually 0000, the second two bytes are the length of the code and are usually 03FB for 127 characters (a five-sector file) or 0800 for 256 characters (a nine-sector file). The last two bytes indicates the address where the file is to be loaded and is usually 07FA.

I always place CHARA1FIX on a separate work disk and then copy the screen font which I wish to work on to this disk. I then make a copy of the font file and name it CHARA1. To check whether the header needs changing, use DISKREVIEW, mark the file and press V. The first few lines of the file will be displayed in hexadecimal format. If the header needs changing, use the Inspect function of FWB by returning to DISKREVIEW and pressing I. By accessing the first sector of the file, necessary changes can be made to the header as described above. If you have never done this before, you can see why using a work disk is recommended. After using CHARA1FIX to amend screen fonts you can reverse the process to restore the header used by FWB.

One final point to remember is that, although screen fonts can be eight pixels wide, which is fine if BASIC or Extended BASIC is being used because screen width is 32 characters, TI-Writer programs operate in TEXT mode, which is 40 characters wide, and a character width of six pixels is necessary. Do not use the two columns on the right of the character display. CHARA1FIX can be used to change the screen appearance of any character from CHR$(0) to CHR$(254).

The TI99/4A typewriter

The following article has appeared in several newsletters. It was written by Charles Ball.

There are often times when we just want to type a short note or letter and, rather than load in a full-blown word processing program, we settle for writing with such low-tech implements as pens and pencils.

It is easy to turn your printer into an electric typewriter. Four lines of BASIC code will do it.

This program enables the user to type a line of text, edit it as desired, and then print it by hitting the Enter key.

Whenever a line of text is to be indented or contains a comma, that line must begin and end with quotation marks. The quotes will not be printed, nor will they be counted in the length of the line of text.

The skip a line, just hit Enter.

The short program No. 1 allows sending of printer codes directly to an Epson RX-80 printer.

By adding a few more lines, the program can be made even more useful. We can require an input as to the maximum width that is to be printed and use this information to set equal right and left margins. A check has been added to ensure that the maximum line length is not exceeded, and it includes a prompt to display what an overly long line can be shortened to. User instructions have also been added. The expanded 10-line BASIC program is listed as program No. 2.

When typing notes, etc., where it is desireable to start printing at column one, input a line length of 80 and monitor the screen for your line length.

A simple way to use this program for correspondence is to use a line length of 56. This will fill exactly two lines of the TI screen. Right margin justification can be accomplished by inserting spaces between words until the second line of text is completely filled.

Of course, the OPEN statement in line four should be changed as required for the printer you are using. The line length feature is designed for pica print. Line three can be changed to accommodate elite or condensed type styles.

NO.1

```
1 OPEN #1:"PIO"
2 INPUT A$
3 PRINT #1:A$
4 GOTO 2
```

NO. 2

```
1 PRINT : : : :"TO INDENT TE
XT OR USE A     COMMA, BEGIN
 & END THAT LINEWITH QUOTATI
ON MARKS": :
2 INPUT "PRESS ENTER TO SKIP
 A LINE. HOW WIDE?(80 CHARAC
TERS MAX)":WIDTH
3 MARGIN=INT((80-WIDTH)/2)
4 OPEN #1:"PIO"
5 INPUT "INPUT A LINE OF TEX
T:       ":TEXT$
6 IF LEN(TEXT$)>WIDTH THEN 7
 ELSE 9
7 PRINT :"LINE TOO LONG! SHO
RTEN TO": :WIDTH;"CHARACTERS
 MAX.": :SEG$(TEXT$,1,WIDTH)
8 GOTO 5
9 PRINT #1:TAB(MARGIN);TEXT$
10 GOTO 5
```

Long lines with text editors

This comes from Bruce Harrison, of Hyattsville, Maryland. He writes:

Here's an item for the User Notes column. This trick was devised in response to a plea from Martin Zeddies of Wolfsburg, Germany.

In a recent letter from a loyal Geneve owner in Germany, there was a plea for some simple way to make TI-Writer (or Funnelweb Text Edit) allow more than 80 characters per line on the printer's condensed mode. The following method will work with most dot-matrix printers, but not all, so a simple test is recommended.

First, through whatever method is needed, put in the control sequence to place the printer in condensed mode. This will allow 136 characters per line on the printer, but the trick is to avoid the carriage return

(See Page 38)

Fig. 1

Here is a sample of stuff being printed in the condensed mode, with nearly 136 characters on each printed line. We can make this keep on going in the manner for the rest of the document if need be, or stop and do something else if that is desired.

Now here's an example of elite condensed, where we can get to the airvana of 160 characters on each printed line, so each two lines on the screen text equals one very long line on the paper. These limits of course do not allow any margins on either side of the paper so the numbers and placement of the central characters would have to be adjusted somewhat to allow for margins. Note the less, this makes for a good tool to save paper, provided the reader's eyesight is good enough.

USER NOTES

(Continued from Page 37)

and linefeed after each 80 characters when printing.

Each line of 136 characters on the paper will occupy 1.7 lines in the TI-Writer document. Thus you'd alternate between lines of 80 and lines of 56 characters length.

Type to near the end of the first line, then hit CTRL-U. Now hit Shift-S, then Enter to get to the next line, then Shift-Q, then CTRL-U to get back into normal typing. On this even-numbered line type only 56 or fewer characters, then a carriage return. The Shift-S while in CTRL-U mode puts a CHR$(19) (Hex 13) in the document, while the Shift-Q puts a CHR$(17) (Hex 11) at the start of the next line. On most dot-matrix printers, that Hex 13 will put the printer off line, so it won't recognize anything except the Hex 11 control character. Thus the printer will ignore the carriage return and linefeed sent by the formatter, and will continue printing with whatever comes after the Hex 11 on the same printing line.

Just continue in this manner with alternating long and short lines for as long as you want to stay in condensed mode with lots of characters per line. This method will work on most models of Star Micronics, Epson, and Panasonic printers. It won't work on some older models, most notably the Epson RX-80, which does not recognize the off-line and on-line control characters. On some printers, you can use a combination of elite and condensed modes to get 160 characters on each printed line, using the same trick to avoid the unwanted CR/LF actions.

See Fig. 1 for a sample made using Funnelweb's Text Edit, and printed with a Canon BJ-200e printer.

Re-mapping the keyboard; the effect of variable names on speed

The following was written by Jim Swedlow and originally appeared as part of his column XB Tips in the California ROM, the newsletter of the User Group of Orange County.—Ed.

You normally see CALL KEY(0,K,S). There are five other values for the first variable, the key unit. They remap your keyboard:

0 — Keeps the keyboard in the same mode as the last time a CALL KEY was executed. If this is the key unit on the first CALL KEY in a program, you stay in 99/4A mode.

1 — Splits the keyboard into two.

2 — Small boards. Good for games.

(See Page 39)

MICROpendium disks, etc.

- ❑ **Series 1996-1997** (May/June 1996-Jan/Feb. 1997, 6 disks, mailed bimonthly) $25.00
- ❑ **Series 1995-1996** (April 1995-Mar. 1996, 6 disks)$25.00
- ❑ **Series 1994-1995** (April 1994-Mar 1994, 6 disks) $25.00
- ❑ **Series 1993-1994** (April 1993-Mar 1994, 6 disks)$25.00
- ❑ **Series 1992-1993** (Apr 1992-Mar 1993, 6 disks).. $25.00
- ❑ **Series 1991-1992** (Apr 1991-Mar 1992, 6 disks).. $25.00
- ❑ **Series 1990-1991** (Apr 1990-Mar 1991, 6 disks) ..$25.00
- ❑ **Series 1989-1990** (Apr 1989-Mar 1991, 6 disks) ..$25.00
- ❑ **Series 1988-1989** (Apr 1988-Mar 1989, 6 disks)...$25.00
- ❑ **110 Subprograms** (Jerry Stern's collection of 110 XB subprograms, 1 disk) $6.00
- ❑ **TI-Forth** (2 disks, req. 32K, E/A, no docs).....$6.00
- ❑ **TI-Forth Docs** (2 disks, D/V80 files) $6.00
- ❑ **1988 updates of TI-Writer, Multiplan & SBUG** (2 disks) $6.00
- ❑ **Disk of programs** from any one issue of MICROpendium between April 1988 and present $5.00
- ❑ **CHECKSUM and CHECK** programs from October 1987 issue (includes docs as D/V 80 file) $4.00

Name _____

Address _____

City _____

State _____ ZIP _____

Texas residents add 7.75% sales tax.. Credit card orders add 5%.
Check box for each item ordered and enter total amount here:

Check/MO Visa M/C
(Circle method of payment)

Credit Card # _____

Exp. Date _____

Signature _____

USER NOTES

(Continued from Page 38)

— Remaps the keyboard as a 99/4 (no A). CTRL and lowercase are inactive.

4 — Pascal mode.

5 — Remaps to the 99/4A mode with lowercase and CTRL active.

When you use 3, regardless of the position of the shift and alpha lock keys, all alpha characters return as uppercase. The problem is that this condition continues until you do a CALL KEY with a different key unit. Try this:

```
10 CALL KEY(3,A,B) :: PRINT : :"Key Unit is 3"
20 INPUT A$ :: PRINT A$
30 CALL KEY(5,A,B) :: PRINT : :"Key Unit is 5"
40 INPUT A$ :: PRINT A$ :: GOTO 10
```

With alpha lock up, try inputting lowercase letters and see what happens.

A key unit of 3 is very useful to make sure that only uppercase characters are caught by the CALL KEY. However, if want lowercase later in an INPUT or CEPT, you must remap the keyboard with another CALL KEY.

VARIABLE NAMES

Longer variable names slow program execution. I ran the following program with progressively longer variable names substitued for <C>:

```
10 C=0 :: FOR I=1 TO 1000 :: A=C :: NEXT I
```

I ran each three times and averaged the results. Here is what I found:

Length of substitute for <C>	Avg. Runtime (seconds)
1	7.74
2	7.72
3	7.76
4	7.9
5	8.04
6	8.12
7	8.23
8	8.34
9	8.46
10	8.56
11	8.62
12	8.71
13	8.83
14	8.94
15	9.02

As you can see, variables up to three characters in length ran in about the same time. Once the length was longer, however, each additional character in the variable name increased the run time by about one-tenth of a second for 1000 executions, or 0.1 millisecond for one.

I also ran this one:

```
10 C=0 :: FOR I=1 TO 1000 :: A=0 :: NEXT I
```

The average run time was 7.06 seconds. There is a cost when substituting variables for numbers. TANSTAAFL (there is no such thing as a free lunch)!

A simple printer trick

The following, which has been edited to fit, was written by Jack Sughrue and appeared in various user group newsletters.—Ed.

I was up early one Thanksgiving Day on my computer, trying to debug a lengthy program I had typed in from a magazine. The program was printed 40 columns wide. My screen view was 28 columns, and my printed out version was 80 columns. I OLDed up the program and entered this one liner with my trusty Gemini turned on:

```
OPEN #1:"PIO" :: PRINT #1:CHR$(27);CHR$(81);CHR$(40)
```

Then I entered LIST "PIO" and, voila, my printout was exactly as the printout in the magazine. I could quickly check all the line endings to see if they matched. I had the program debugged within a half hour.

People I've shared this with have felt the same way. It was basically a matter of looking through my printer manual and noticing the right margin code: 81. The rest is self-explanatory. I've also used this for printing 28-column program listings for use in newsletters and articles.

MICROpendium pays $10 for items used in this column that are sent in by readers. Send User Notes to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680; e-mail jkoloen@io.com.

CLASSIFIED

FOR SALE

FOR SALE

TI Hexbus CC40 Computer $55 and Hexbus RS232 $60 (both mint condition). Zenith composite monitor $40. Call Tom at (703) 455-9216. v13n3

HALF PRICE W/DESK

lellent condition, complete TI99/4A system, RS232 card, 2 disk drives, original color monitor, programs, Multiplan, TI-Writer, record keeping games, MUCH MORE. Paid over $400, honest business owner, (Rod) 512-244-6064 AFTER 6 P.M. v13n3

FOR SALE

TI99/4A computer, TI-Writer Word Processor Program, 42 issues MICROpendium June'92-December'95, 2 games, 6 management programs, 60 disk programs, 2 floppy disc drives w/32K memory. MAKE OFFER. George Reier, 111 Loquat Rd., N.W.; Lake Placid, Fl.; 33852; (941) 465-3624 v13n3

Call 512-255-1512 to place a classified ad. The cost is 10 cents per word.

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**

❑ 12 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 appears on your mailing label when making an address change.

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

Check/MO ❑ MasterCard ❑ VISA ❑ (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%.

Mail to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680

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 shipping 75 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