Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 4 May 1995 $3.50

LIMA

Super AMS • SCSI Card • Funnelweb Editor • Term 80 • RXBV.1001 • PC99

English and all the languages we humans use have large enough vocabularies to allow us to say virtually anything we want, but yet have enough inconsistency that systematic computer translation of languages is a very difficult proposition for reasons that, if they are not already obvious to you, soon enough will be.

Translation

See page 6

- √ Call files from XB in assembly
- √ Another XBASIC Tic-Tac-Toe game
- √ Microcom Networking Protocol
- √ Reviews of PFM+, Board Games

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published monthly 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.

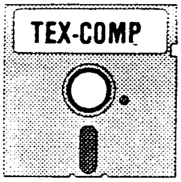
Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone & FAX: (512) 255-1512
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@io.com
John Koloen..............................Publisher
Laura Burns..................................Editor

*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.



ANNOUNCING NEW LOW PRICES ON FREEWARE DISKS

FROM

$2.95 TEX COMP LTD. $2.95

TEX-COMP HAS SLASHED PRICES ON THE BIGGEST & BEST COLLECTION OF FREEWARE FOR THE 99/4A. NOW ONLY $2.95 PER DISK WITH A 5 DISK MINIMUM. CHOOSE FROM HUNDREDS OF GREAT PROGRAMS. NO EXTRA CHARGE ON PROGRAMS THAT REQUIRE MORE THAN ONE DISK SIDE. ALL PROGRAMS HAVE BEEN TESTED AND ALMOST ALL HAVE BEEN PROVIDED WITH EXBASIC AUTOLOAD. CHOOSE FROM THE BEST IN GAMES, UTILITIES, GRAPHICS, HOME & BUSINESS AND DISK BACKUPS OF DOZENS OF YOUR FAVORITE MODULES THAT ARE NOW OUT OF PRODUCTION.

GAMES • BUSINESS • GRAPHICS • WORD PROCESSING • UTILITIES • DATABASE • MUSIC • COMMUNICATIONS • HOME

GRAPHICS, MUSIC & ANIMATION

#1 THE SINGING TI VOL. 1 (S)(2)
#4 PRINTART (2)(P)
#5A MUSIC/GRAPHICS
#6 EXBASIC MUSIC (2)
#7 SPACE SHUTTLE MUSIC/GRAPHICS
#9 MONA LISA PRINTOUT (P)
#11 ANIMATED XMAS (WOODSTOCK)
#14 FIGURE STUDIES (P)(PG)
#32 EXBASIC XMAS MUSIC (2)
#41 VIDEO GRAPHS (M)
#52 ANIMATION 99' (2)
#69 PLAYER PIANO/KEYBOARD ANALYSIS
#93 KBGB GIRLIE CALENDAR (4)(P)
#103 SORGAN THE TI ORGAN
#107 STARTREK MUSIC ALBUM
#111 POP MUSIC & GRAPHICS
#114 PANORAMA
#115 GRAPHICS DESIGN SYSTEM
#120 BITMAC (2)(P)
#230 THE SINGING TI VOL. 2 (S)(2)
#231 THE SINGING TI VOL. 3 (S)(2)
#246 THE SINGING TI VOL. 4 (S)(2)
#313 3-D WORLD (CAD for the TI)

COMPUTER UTILITIES, PRINTER UTILITIES & PROGRAMMING LANGUAGES

#3 DUMPIT (E/A)
#15 STAR/EPSON PRINTER DEMO (P)(2)
#16 SIDEWAYS PRINTOUT (P)(2)
#18 TI DIAGNOSTIC (MM)(2)
#28 LOADERS & CATALOGERS
#30 HOUSEHOLD BUDGET PRINTOUT (P)
#35 PROGRAMMING AIDS & UTILITIES I
#42 FUNNELWEB FARM (SHELL UTILITY)
#53 HACKER CRACKER
#55 SCREEN DUMP (P)
#62 DISK MANAGER II (M)
#75 DISK CATALOGER
#76 PROGRAMMING AIDS & UTILITIES II
#78 ARTCON GRAPHIC CONVERSION
#79 DISK MANAGER 1000
#80 BIRDWELL DISK UTILITY (2)
#85 AUTOBOOT UTILITY
#86 COLUMN TEXT III (P)
#87 ARCHIVER III
#89 PROCALC DECIMAL/HEX CONVERTER
#96 STATISC & SORT ROUTINES
#97 MEMORY MANIPULATOR
#101 ENHANCED DISPLAY PACKAGE
#108 FUNLPLUS (PLUS!)
#110 DISK+ AID
#117 UNIVERSAL DISASSEMBLER
#119 RAG LINKER CONVERSION
#127 PIX GRAPHICS UTILITY
#243 OS/99 (GD)
#251 PC TRANSFER TI/IBM (DD)
#253 THE EXPLORER/DM1000
#254 NIBBLER/TURBO
#260 TI FORTH (DISK ONLY- add $8 for manual)
#17 TI FORTH DEMO
#116 TI FORTH TUTORIAL
#5079 FORTH SOURCE CODE
#104 C99 COMPLIER & LIBRARY
#5007 TEACH YOURSELF TI BASIC
#5019 TEACH TOURSELF EX-BASIC
#5067 BEGINNING BASIC TUTOR
#307 GRAPHICS CODE GENERATOR
#3912 SUPER BUGGER
#3913 BETTER BANNERS
#3914 CERTIFICATE 99
#3915 HOROSCOPE MAKER
#3916 GRAPHX+ PRINT SHOPPE
#3917-3720
GRAPHX COMPANIONS 1-4
#3721 MAC FLICK (G)
#3722 PRINTING TO GO (G)
#3723 GRAPHX DINOSAURS (G)

BUSINESS, ACCOUNTING, WORD PROCESSING, DATA BASE HOME OFFICE & HOME

#10 GOTHIC PRINTOUT (P)
#19 TI WRITER/MULTIPLAN UPGRADE
#20 ACCOUNTS RECEIVABLE (2)
#21 DATA BASE DEMO
#23 WILL WRITER
#29 LABEL MAKER I (P)
#36 STRICTLY BUSINESS (2)
#56 SPREAD SHEET
#58 PR BASE (database)
#59 GRAPH MAKER
#74 LABEL MAKER II (P)
#77 MICROdex 99 (database)
#81 HOME ACCOUNTING SYSTEM
#83 HOME APPLICATION PROGRAMS (2)
#90 JET CHECKBOOK MANAGER
#92 HOUSEHOLD INVENTORY
#109 TI WRITER MINI MANUAL
#112 INVOICE PACK
#113 LABEL MAKER III
#129 CASH DRAWER (point of sale)
#130 THE ORGANIZER
#147 CALENDAR-NOTEPAD
#177 HOUSEHOLD BUDGET MANAGEMENT (M)
#30 HBM DATA PRINTOUT (P)
#221 PERSONAL REAL ESTATE (M)
#249 MAPMAKER
#252 99 WRITER II (TI WRITER)(P)
#253 AMA MAILING LIST
#255 TRAK-A-CHECK
#257 DAILY DIARY
#257A EXER-LOG
#23 WILL WRITER
#310 SELP HELP TAX CUT

TELECOMMUNICATIONS (MODEM)

#57 TELCO
#57D TELCO (FOR SYSTEMS WITH DS DRIVES)
#118 FAST TERM

EDUCATION & PERSONAL DEVELOPMENT

#22 ASTROLOGY
#24 ENGINEERING CALCULATIONS (2)
#25 MEDICAL ALERT
#27 KIDS LEARNING I (2)
#31 MORSE CODE TRAINER
#37 LAPD COOKBOOK (2)
#40 ARTIFICIAL INTELLIGENCE (ELIZA)
#54 ASTRONOMY
#66 HEBREW TYPEWRITER
#67 GENEALOGY (2)
#71 KIDS LEARNING II (2)
#95 WEATHER FORECASTER +
#138 FIREHOUSE COOKBOOK
#142 TOUCH TYPING TUTOR (M)
#184 FACE MAKER
#194 ST. NICK/GHOSTLY SPELLING (M)
#195 TINY LOGO
#199 MILLIKEN ADDITION (M)
#200 MILLIKEN DECIMALS (M)
#203 MILLIKEN FRACTIONS (M)
#204 MILLIKEN INTEGERS (M)
#205 MILLIKEN LAWS OF MATH (M)
#211 MIND CHALLENGERS (M)
#212 MINUS MISSION (M)
#213 MILLIKEN PERCENTS (M)
#214 STORY MACHINE (M)
#215 BEGINNING GRAMMAR (M)
#216 METEOR MULTIPLICATION (M)
#217 HANGMAN (M)
#222 MUSICMAKER (M)
#223 PHYSICAL FITNESS (M)
#225 ALIEN ADDITION (M)
#226 ALLIGATOR MIX (M)
#227 DEMOLITION DIVISION (M)
#228 DRAGON MIX (M)
#270 TRIGONOMETRY
#271 CALCULATORS & CONVERSIONS (2)
#272 HIGHER MATH (2)
#273 ASTRONOMY 2
#306 SPEAK & SPELL II (S)(EX)

GAMES

#2 WH. OF FORTUNE, BLKJACK, POKER
#3 TI TREK
#8 LOTTO PICKER
#13 STRIP POKER (PG)
#26 R-RATED NOVELTY GAME
#33 CHECKERS & BACKGAMMON
#34 SOLITAIRE & SCRABBLE
#38 GREAT TI GAMES VOL 1
#39 GREAT TI GAMES VOL 2
#43 BEST OF BRITAIN GAMES VOL 1
#45 BEST OF BRITIAN GAMES VOL 2 (LEGEND OF CARFAX ABBY GRAPHIC-INTERACTIVE ADVENTURE)
#46 SUPER TRIVIA 99
#47 INFOCOM RAPID LOADER
#48 GHOSTMAN (from U.K.)
#49 DEMON DESTROYER (from FRANCE)
#50 OH MUMMY! (HIT from GERMANY)
#51 BERLIN WALL (from CANADA)
#50 FREDDY (HIT from GERMANY)
#51 THE MINE (from GERMANY)
#53 ASTROBLITZ & MAZOG
#54 MAJOR TOM & SPACE STATION PHETA
#55 PERFECT PUSH (HIT)
#58 CHESS (SARGON)
#70 TI RUNNER II (HIT)
#72 CEBERUS (HIT SPACE GAME)
#73 CRYTO (GRAM)
#82 CROSSWORD (PUZZLES)
#84 GALACTIC BATTLE & SPY ADVEN.
#88 AUSSIE GAME COLLECTION VOL 1
#91 THE MAZE OF GROG (WOODSTOCK II)
#94 GREAT TI GAMES VOL 3
#98 DAYS/DOORS OF EDEN (BIBLE ADV)(AM)
#99 GREAT TI GAMES VOL 4
#100 ASSAULT THE CITY (TD)
#102 COLOSSAL CAVES (ADVENTURE)
#105 KINGS CASTLE (M)
#106 QUEST (D&D)
#121 SUPER YAHTZEE & WHEEL II
#122 ADULT ADVENT & GAMES (PG)
#123 GREAT TI GAMES VOL 5 (2)
#124 GREAT TI GAMES VOL 6 (2)
#125 BLACKJACK & POKER (M)
#126 VIDEO CHESS (M)
#128 TETRIS (HIT from RUSSIA)
#131 COMPUTER CRAPS
#132 AMBULANCE (M)
#133 DRIVING DEMON (M)
#134 ROTO-RAIDER (M)
#135 ARTURUS (HIT-ZARGON)
#136 ANT-EATER (M)
#137 CROSS-FIRE (M)
#139 MOONMINE (M)
#140 MASH (M)
#141 MOONSWEEPER (M)
#143 CONGO BONGO (M)
#144 STAR TREK (M)
#145 BUCK ROGERS (M)
#148 KENO & SLOTS
#149 GREAT TI GAMES VOL 7 (with HIT BLOCKBUSTER)
#150 ULTIMATE TRIVIA
#151 JUNGLE HUNT (M)
#152 POLE POSITION (M)
#153 DONKEY KONG (M)
#154 PROTECTOR II (M)
#155 PAC MAN (M)
#156 CENTIPEDE (M)
#157 DEFENDER (M)
#158 SHAMUS (M)
#159 MS. PAC MAN (M)
#160 DIG-DUG (M)
#151 PICNIC PARANOIA (M)
#162 MOON PATROL (M)
#3711 ARCADE SPECIAL (4 GAMES)
#3712 THREE GREAT GAMES
#3713 PRO TENNIS+

INFOCOM ADVENTURE BACKUPS

#163 ZORK I
#154 ZORK II
#155 ZORK III
#156 HITCHIKER'S GUIDE
#167 WITNESS
#168 ENCHANTER
#169 INFIDEL
#170 PLANETFALL
#171 SORCERER
#172 DEADLINE
#173 CUTTHROATS
#174 SUSPENDED
#175 STARCROSS

****

#176 AMAZING (M)
#178 DEMON DESTROYER (M)
#179 POPEYE (M)
#180 QUEBERT (M)
#181 METEOR BELT (M)
#182 BLASTO (M)
#183 CAR WARS (M)
#184 FACE MAKER (M)
#185 SUPER FLY (M)
#186 SPACE BANDITS (M)
#188 KILLER CATERPILLER (M)
#190 BLACK HOLE & SPACE AGRESS. (M)
#191 GREAT TI GAMES VOL 8
#192 GREAT TI GAMES VOL 9
#193 SPY'S DEMISE (HIT) (M)
#194 ST. NICK+ (M)
#196 JOTTO
#197 PRO TENNIS+ (HIT)(M)
#198 TI INVADERS/TOMBSTONE CITY (M)
#202 CONNECT FOUR (M)
#205 HOPPER (M)
#205 TREASURE ISLAND (M)
#206B SLYMOIDS (M)
#207 OTHELLO (M)
#208 PARSEC (M)
#209 SOCCER (M)
#210 SEWERMANIA (M)
#218 HUSTLE/FOOTBALL (M)
#219 CHISOLM TRAIL (M)
#220 ZERO ZAP (M)
#224 ATTACK (M)
#229 4A FLYER (FLIGHT SIM.)(M)
#232 TUNNELS OF DOOM (MOD BACKUP PLUS 2 NEW ADVENTURES)(M)
#233 MS ADVENTURES (3 ADV-EXBASIC)
#248 STRIKE THREE BASEBALL (M)
#248 GREAT TI GAMES VOL 10
#250 BARRAGE/SPOTSHOT
#318 THREE GREAT GAMES
#319 ARCADE SPECIAL
#317 BEANSTALK ADVENTURE

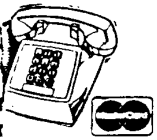
****

SCOTT ADAMS ADVENTURES
(USE WITH ADVENTURE MODULE)
#349 TI ADVENTURES 1-13+
#350 TI ADVENTURES 14-16+

P = Printer required
G = Graphx required
S = Speech required
M = Module Backup
MM = Mini Memory req.
E/A = Editor Assem.
Exbasic and 32k mem req. for most programs.

Authorized Dealer

ORDER BY PHONE



(818) 339-8924 FAX (818) 858-2785

24 HOURS A DAY

VISA

TEX COMP LTD.

America's Number One TI computer retailer

425 E. Arrow Hwy. #732
Glendora, CA 91740-5684

TERMS: MINIMUM ORDER FOR $2.95 PRICE IS 5 DISKS (REG. 4.95) ADD $4.00 PER ORDER FOR SHIPPING (U.S.) ALL PRICES ARE FOR CASH/CHECK, ADD 3% FOR CREDIT CARD ORDERS. INCLUDE STREET ADDRESS FOR U.P.S.

COMMENTS

Lima fair scores a hit

The Lima Multi User Group Conference looks to have been another success. Attendance was well over 100 and, judging from the variety of workshops, there was more than enough going on to keep TIers busy. Charles Good, the person in charge of the Lima fair, deserves a tremendous amount of credit for making this event something that TIers look forward to every year.

The big news from this year's fair concerned products developed or marketed by the defunct Asgard Software. First, the Super AMS card is now being supported and produced by the Southwest 99er user group. A lot of time and resources went into the development of this card and it's terrific that it will continue to be available. The card gives users of the 99/4A up to 256K of bank-switched memory.

Other Asgard products will continue to be marketed as RAMcharged Computers has purchased Asgard's inventory. Asgard marketed a lot of programs for the TI, and this deal can only benefit TI users.

For more information about other products demonstrated at the fair, see page 5.

SCSI PROGRESS

Although progress on the TI version of the Western Horizon Technologies SCSI card was evident in a demonstration at the Lima fair, it still supports only several basic commands. While work progresses on this version, I have nothing to report on my system. As you may know, I have a SCSI connected to my Geneve. Unfortunately, the SCSI doesn't work with the MEMEX memory expansion card installed in my system. I've been told that the incompatibility can be fixed. I'm hoping it can, though I have my doubts. I don't look forward to choosing between the MEMEX and the SCSI. Both of them are expensive and useful. Because the MEMEX required a number of modifications to the Geneve card, I'm reluctant to undo the work. Simply pulling the MEMEX out of the PEB and adding a 32K chip to the Geneve doesn't help. I definitely would like to have my cake and eat it too.

CHICAGO FAIR UPDATE

Hal Shanafield reports that plans for the Chicago TI Fair are well underway. The fair will be held October 28 at the Evanston Public Library. Deadline for vendor registration is Sept. 1

—JK

FEEDBACK

Staying with it

Thank you for your continued support of the TI community by publishing MICROpendium. I look forward to each issue. I don't understand programming, so some articles are of no interest to me. I do appreciate the Reviews, Comments, Feedback and Tips sections. I'm a diehard TI user, and as long as you publish MICROpendium, I'll subscribe.

Wally Scheele
Sheboygan, Wisconsin

Thanks for response

I just wanted the readers of MICROpendium to know that I received several speedy replies to the January 1995 "Reader to Reader" asking for copies of NAME-IT and the RMJ Mailing List to use in compiling a database of 99/4A and 9640 users. Specific thanks to Gary Fitzgerald, Harold Oldroyd and Robert Clark, the first 99ers to come through with the programs requested I would also like to thank Harley Ryan Jr. and Dave Renkenberger for their generous offers to help the "cause."

When I called for help on mailing list management software, I was generally, but not specifically, familiar with the programs requested and was already in the process of writing my own program in case one of the requested programs just didn't "fit" the needs of the name and address database project. Some readers may wonder why in one issue of MICROpendium I was asking readers to send me copies of mail list management software and in the next issue I was releasing my own mail management program. The reason is that by February I had decided to use my own program on the project.

After examining NAME-IT, the RMJ Mailing List, YU-CAN Business Manager and the TI-Count Mail Manager, I found that all had strengths and weaknesses. Oddly enough, most have no provision for addresses outside the United States, meaning that their is no input field for Nation. That shortcoming proved fatal for some of the programs.

Larry Tippitt's Mailroom was the front runner for the project all along, but I even decided against it because I wanted even

(See Page 5)

Lima report

Hardware, software in limelight

The following was posted on the Internet by Charles Good, organizer of the Lima MUG Conference.—Ed.

Here is a short report of the Lima Multi User Group Conference:

One hundred fourteen people signed the sign-in sheet. One hundred twenty nametags were used. I know of several people there who didn't sign the sheet. Some members of the public came in and wandered around without signing in. I think it is safe to claim an attendance figure of 125.

The Super AMS card, 128K or 256K of bank-switched memory for the 99/4A, was shown and orders were taken by the SouthWest 99er user group. A production run of 100 is scheduled for this week. *(For information or to order, contact the SouthWest 99ers, P.O. Box 17831, Tucson, AZ 85731, or call the Cactus Patch BBS, (520) 290-6277.—Ed.)*

Working SCSI hard drive demonstrations were videotaped using systems working on a Geneve and a 99/4A. The /4A right now is limited to OLD, SAVE and DELETE programs from BASIC to and from the hard drive. Now that David Nieters has a working SCSI card, rapid additional progress is expected for 99/4A SCSI. The Geneve version is largely complete, except that there is yet no DSK1 emulation and no way to boot MDOS directly from the SCSI drive. There are advanced disk management and sector editor tools available, and these are shown on the videotapes. *(For information or to order, contact Bud Mills Services,(419) 385-5946 Ed.)*

The latest Funnelweb editor for 80 columns gives you your choice of one 128K text buffer, or two 64K buffers, or four 32K buffers with the ability to cut and paste between buffers.

It is now possible to load the SCSI version of MDOS into and autoboot from a flashdisk on a Geneve. Before the MUG Conference this was not possible. Contact Cecure Electronics about this at (800) 959-9640.

See related article, page 28

I demonstrated TERM 80, which gives you an 80-column display, and real VT100 emulation on an unmodified 40-column 99/4A system. It's on the videotape. The audience was amazed. *(For information/orders, contact Jeffery Brown, 2111 Montreal Rd. #102, Gloucester, Ontario, Canada K1J-8M8.—Ed.)*

RXB V.1001 now has working batch file capability. It is for GRAM devices and is now public domain.

PC99 Stage 3 really blew the audience away. It is very fast if you have a fast computer, and acceptably fast even on a 386. This is a 99/4A emulator that runs on an IBM compatible. *(For information/orders, contact Cadd Electronics, 45 Centerville Dr., Salem, NH 03079, (603) 893-1450.—Ed.)*

RAMcharged Computers has purchased the entire inventory of Asgard software from Harry Brashear. These are all legal copies and will be available to the TI community as long as supplies last, which may be years since this is a very large inventory. Call RAMcharged at (800) 669-1214.

In summary, significant new hardware and software products were shown. There is life in the old 99/4A yet!

FEEDBACK

(Continued from Page 4)

more features in the program than any existing mailing list application offered. For instance, MLM (my mailing list manager) can generate TI-Writer Value Files, address envelopes, generate one-time address labels from info entered on the screen for quick addressing, print warning labels and read the contents of a diskette and print a label for it. MLM provides contact management capabilities so a user can keep track of correspondence with clients and other users, and more. No existing application put all those things together along with all the other features mentioned in the February 1995 MICROpendium newsbyte.

I received a tentative offer to put the database together using TC-Mail, but I realized that the software needed to be universally available, another reason I went with MLM, as I can distribute it. Not everyone has TI-Base, and, frankly, I'm not sure where it can be purchased anymore. MLM is shareware, which means I encourage duplication and distribution among the TI community. As long as I am an active member of the TI community, MLM will be available and I will support it.

I have sent out more than 60 copies of MLM to Canada, England, Germany, Italy, Sweden and the United States. I am looking for active 99ers (users groups, actually) in Africa, South America and the Far East who would like a copy, as well as users groups in the Pacific Northwest of the United States. For a copy of MLM for your club's library, please send two formatted single-sided, single-density floppies with return postage and a mailer to William Gaskill, 2310 Cypress Court, Grand Junction, CO 81506 USA. If you don't mind having the program in archived format, send one preformatted SSSD floppy, and I will archive the programs using Barry Boone's Archiver 3.03.

This international resource will henceforth be referred to as the Int'l 99er Mailing List. If you or the members of your users group would like to be included in the list I would like to hear from you. The MLM program has input fields to record First Name, Last Name, Address, City, State/Province, Postal Code, Nation, Group Code (a user-defined use), Home Phone, Work Phone, Fax/BBS Number, Dates, 56 characters of Notes and a Contact Management feature allowing up to 114 lines of free-form text to be associated with a record via a separate file. Send me info for any of the available fields and I will make it part of the record.

I would like to offer my sincerest thanks to Dr. Charles Good of the Lima Users Group in Lima, Ohio, and Ralph Rees of the VAST Users Group in Phoenix, Arizona, for their support

Bill Gaskill
Grand Junction, Colorado

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

Language translation by computer

By STEVEN L. RICHARDSON

From all appearances, languages aren't that hard to learn; nearly everybody in every country in the world becomes fluent in at least one by the time of his or her third birthday. A second language becomes a little harder. Perhaps it's because our brains only have room for one image of an apple and its accompanying word, and have difficulty associating the image with two words at once, though I know people who were raised in bilingual homes who can speak and seemingly think in both languages.

I'm told that by the time you've learned five languages, the next five are learned very quickly. This may be true, but my studies in college of a second language (German) convinced me that the biggest obstacle to becoming multilingual is that second language.

About 15 years ago I became interested in the history of the development of a technology called photographic surveying (now more commonly known as photogrammetry), whereby maps are constructed from photographs taken today with the camera pointed earthward from aircraft and satellites, but in the 19th century the photographs were taken usually horizontally from mountain tops. There were American pioneers in this field, among them were Henry Reed, John Flemer, James Bagley, and Charles Wright; earlier there was Canada's Surveyor General, Edouard Gaston Daniel Deville. I discovered that much of Deville's professional correspondence is preserved at the Public Archives Canada in Ottawa and eagerly bought a plane ticket to go there and look at the collection. When I arrived I was pleased to see that Deville spoke and wrote fluent English. Unfortunately, since he was born in France, he also spoke and wrote fluent French. Worse, I found a file of 500 or more pages of correspondence in French between Deville and Aimé Laussedat, who invented photographic surveying in France beginning about 1849. The pages were curled together in such a way that it was obvious that they had not been separated to be read for many decades. The curator of Record Group 88 was a friendly young man with a beard. I asked him if there was some way that I, not understanding French, could somehow get some idea of what Laussedat and Deville wrote about to each other in the early 1890's. He smiled and loaned me a black book about three inches thick. I opened it to discover a French-English dictionary.

English and French share enough similarities so that here and there we find a word in French that resembles an English one. Most of these similarities stem from contacts (friendly and unfriendly) the English had with the French people in the past, as well as conquest of each group by the Romans 2000 years ago. Looking at a page of French I can kind of tell here and there what is being discussed, but not really what is being said. Supposedly, a few hours of work (per page) with the French-English dictionary could shed more light on that, but I never had that kind of patience. A few years later a Texas Instruments 99/4A computer found its way into my home, and I discovered that it does have that kind of patience, as well as hidden talents that could only be brought out by giving it the proper instructions.

Did you ever try to talk to someone in BASIC? The language is wonderfully consistent, TI users in Germany and France and in fact all over the world use it. Although it resembles English, it has an extremely limited vocabulary. English and all the languages we humans use have large enough vocabularies to allow us to say virtually anything we want, but yet have enough inconsistency that systematic computer translation of languages is a very difficult proposition for reasons that, if they are not already obvious to you, soon enough will be.

I wrote a program called TRANSLATOR to help me deal with my dilemma; a copy of the code is provided with this article. If you choose to use it to translate a document, this is how your computer will translate: You type in the foreign text, the program looks through the text, compares each word to its own list of words which begin with that letter and substitutes an English word for each foreign word it recognizes. If the word isn't on the list, it is left untranslated.

The list of words it compares against will begin on line 5000. Because I have no idea which language you are interested in translating, or whether you will be translating love letters or legal documents, I will leave it to you to create a set of DATA to suit your needs. A utility program called DATACREATE is provided to assist you in this task, and it will be explained later.

Since neither of the above programs knows any language beyond BASIC, you will at some point need to go shopping for a suitable foreign language dictionary to use in teaching it. Here the most expensive or extensive is almost certainly not what you need. Your computer will be substituting one English word for one foreign word, so the best dictionary will be one that tries whenever possible to do the same thing, without expecting you to know how to conjugate verbs or understand grammar. The real translation will be taking place in your mind, as you study the results of your computer's research. Your machine's memory has limits, of course, so the best dictionary might be found at the back of a phrase book or beginning foreign language textbook, such as you should be able to find at a local thrift store.

As the program is designed, it is capable of working with any language that uses the Roman alphabet. Without too much difficulty, by redefining the graphics, you can include special characters as needed, even to the entire Greek or Russian alphabets, but it will be best to restrict this sort of activity to a minimum — at least until you know better how the program works, and whether you like the kind of work it does. Many modern printers are capable of redefining graphics in their RAM to accommodate a new alphabet, but you would also need to learn to type in that alphabet.

After the program TRANSLATOR has been loaded type LIST and note, below the header, the following statements:

```
150 W=831
160 DIM A$(831), B$(831)
```

(See Page 7)

TRANSLATOR —

(Continued from Page 6)

The value of W and the DIMensioning of A$() and B$() set the limits of your file. Changing the value for W and DIM changes the maximum number of words you can process at one time. Increasing the number allows you to input more words, but could also mean a memory overflow during processing which would slow the work of the computer down considerably. With the typical expanded system you should be able to teach your computer a vocabulary of around thousand foreign words and their English equivalents. If you have a Geneve, or if hardware ever comes forth to increase your TI's memory capacity, the limits can be moved closer to infinity simply by altering the dimensions defined on lines 150 and 160.

After you load the program and type RUN; the program will display a menu which lists the available options:

```
FOREIGN
LANGUAGE TRANSLATOR
1 CREATE FILE
2 ADD TO FILE
3 LOAD FILE
4 TEXT
5 LIST
6 FIND STRING
7 EDIT
8 SAVE FILE
9 DEFINE BLOCK
10 TRANSLATE
11    ALPHABETIZE
12    VOCABULARY COUNT
13    UNDEFINED WORDS
14 FOREIGN WORD LIST
15 BREAK
LAST CHOICE= 0
0 WDS
SELECT ONE: _
```

You may notice a resemblance of this menu to the STYLOMETRY program that was published in the February 1994 issue of MICROpendium, and in fact the purpose for many of the functions are shared between the two programs. Briefly stated, the menu choices are explained as follows:

1 CREATE FILE first instructs you to release the alpha lock key so that all typing is done in lower case. This way, should you wish to use special European characters, they will not be much different from their unaltered counterparts. The file you will be creating is a sample of foreign language text of 830 words or more. As you begin a session, you are given an opportunity to identify the source document that you are typing from. Next the program prompts you to type a single foreign word followed by ENTER, then type the next word, followed by ENTER, and so on. As you do this you should remember that no word should be capitalized; the only punctuation that the program can deal with is the period and apostrophe. You can keep typing words in until you have either reached the limit of 830 words or else type in "xx" at any time and ENTER to return to the menu.

You will probably prefer to create the foreign text as a text file using a word processor such as TI-WRITER, or download a file of foreign text from a bulletin board. The LOAD FILE option has a way of transforming a document imported in this way to the format used by the TRANSLATOR program. This may also work for foreign language files that have been downloaded using your modem. Simply precede the file name with "=", as "=DSK2.filename", or "=filename" (if no drive is specified, saving and loading goes to DSK1 by default).

Lines 300 to 320 of the TRANSLATOR program include some examples of special European graphics that can be used by pressing SHIFT 2,3,4,...0 (the characters formerly were @@, #, $, %, etc.). These characters can be adapted, rearranged, or eliminated as needed for the specific needs of the language you are going to be dealing with, and identified on a paper strip above the number keys. If there are words that begin with any of these characters, the CALL CHAR statements should be modified so that the character being defined is ASCII 64: "@@", or ASCII 91 to 96: "[\]}_`". You will find that in European languages, most of the special characters are merely cosmetic, being guides on how to correctly pronounce the word. There may be some characters that you will have to use, but not as many as you might think. However, a Vietnamese friend informs me that in his native language, which uses a Roman alphabet, the special characters change the entire meaning of a word, as also the pronunciation, so that "di" means move, "d+" means aunt, "dl" means threaten.

Option 2 permits you to add to the file, letting you pick up where you left off, and even listing the six previous words as a reminder of where that was.

Option 3 LOAD FILE allows you to load a file from disk that has previously been saved (Option 8). The operation takes about 2 minutes for 830 words. As mentioned above, you have also been provided a way by which you can import a text file. If you load in this way it takes 2.5 minutes per 100 words as the file is transformed to fit the requirements of the program. Some editing may be necessary to remove or alter special characters that may have been imported.

Option 4 TEXT allows you to see the words in paragraph form; if you would like to stop the display at any time, press any key, just as you would temporarily stop an Extended BASIC listing. The program lets you choose between English and foreign (assuming the file has already been translated using Option 10). The original text or the translation can also be sent to the printer.

Option 5 LISTs the foreign words and their English equivalents in parallel columns (if Option 10, TRANSLATE has already been run) and also identifies each word by a reference number, which can be used to EDIT words (option 7). If you would like to drop out at some point without seeing the entire list, hold down "Q". As with option 4, pressing any other key can pause the display.

If you need to search for a word that you later notice has been misspelled, or perhaps if you later discover that a special graphic character needs to be substituted for a letter in the word, the word can be fixed using option 6 FIND STRING. The computer will list on the screen all words which contain the letter or combination of letters that you specify. If you have a speech synthesizer attached, the computer reads the word numbers to you aloud, and you can copy the numbers down by hand without having to refer to the screen.

Option 7, EDIT, allows you to change the spelling of a word, make room to Insert a word, or Delete a word as needed. You

(See Page 8)

TRANSLATOR—

(Continued from Page 7)

can also Show a word of your choosing accompanied by the three words preceding and the three words following.

Option 8 SAVE FILE saves what you have typed or downloaded onto disk for later loading and processing.

Option 9 BLOCK lets you select a segment of the data set to work with. Because we are using your computer near the limits of its memory capabilities you should translate only in blocks of about 100 or 200 words at a time, purging before proceeding to the next block. You can also use Option 9 if you want to see only portions of the file in the TEXT, LIST, and FIND STRING options.

Option 10, TRANSLATE, the one you've been waiting for, translates the file word for word, assuming that you have a DATA list MERGEd into the program, beginning at line 5000. It does not conjugate verbs or rearrange grammar. If it can't find an exact equivalent it gives the foreign word between "<>" symbols. Of course some time is consumed as the computer sorts through the DATA list looking for the words it needs, requiring about 5 minutes per 100 words. If you change your mind while translating, hold the "Q" key down through one cycle to return to the menu.

You will be using options 11 to 13 to do research on samples of foreign text. The original order of the samples will be destroyed, so the text files should be saved on disk before selecting one of these. The following options are intended to be used in consecutive order. First, Option 11 ALPHABETIZE sorts all the words into alphabetical order. Option 12 VOCABULARY COUNT counts the number of occurrences of each word from the alphabetized list and displays the results in either alphabetic or numeric order. If you choose numeric you will see the most frequently occurring words at the top of the list, which is probably the preferred choice for most of what you will need to do. Option 13 UNDEFINED WORDS allows you to see which of the words have not yet been defined on the DATA list, after a DATA list has been created and merged into the program. You should be unable to use this option until there is already a DATA list in the program.

Selecting Option 14, FOREIGN WORD LIST will result in either of two things. If there is no foreign text file in the computer, the computer will load the DATA statements into the file and sort the foreign words into alphabetical order, then list them and as either a

"1 Dictionary" — 350 words at a time, accompanied by their English equivalents, or

"2 Word List" — 830 words without translations.

This will allow you to determine if any of the words have been defined more than once, and, by comparison to the published dictionary, whether any of them have been incorrectly defined. If there is already a text file in the computer you are not offered the above choice, but will instead see a list of foreign words and their English equivalents in the order in which they appear in the DATA list.

In summary then, when you are ready to begin, load the program, Type RUN, select option 1, and type in or download your text of 830 foreign words (the computer will return to the menu should you exceed the limit), then save them to disk. Next alphabetize the file. After that do a vocabulary count with the numeric option. Print out the result. When you have compiled a list of undefined words, work your way through the dictionary coming up with (whenever possible) one-word definitions for each. If you need to, the computer will accept short phrases, such as "to be", or multiple meanings separated by a slash, such as "rip/cry" (that's a definition in English of the English word "tear", which though it is spelled one way, it has two different meanings and two different pronunciations; there is a larger body of words in every language that have one spelling, one pronunciation, and multiple meanings — examples can be found on every page of an English dictionary). Care should be taken to ensure that the meaning you select is the one most likely intended by the author of the source you wish to translate. Eliminate words from your list that are already similar to their English equivalents — this way your computer can concentrate its efforts on the words that you are not already likely to recognize.

In English there are certain words we use more frequently than others, such as "and", "the", "of" and "to". These words also have equivalents in other languages. The computer will find those words in a sample of text, allowing you, if you choose, to put them and their definitions at the beginning of the list.

The TRANSLATOR program needs a vocabulary to work from, which has to be read from DATA lists. The DATACREATE program simplifies this part of the task. Experiment with it before you get serious so you understand how it works. All typing you do is with the alpha lock key up. You type a foreign word, a space, and the English equivalent. The program transforms the English word or phrase to uppercase letters, and creates a MERGE file consisting of DATA statements. You should type all the words beginning with the same alphabetic character before passing on to another character. Provided the language you name is the same, files created during later sessions can be merged with your first file, prior to — or even after merging the entire collection to the TRANSLATOR program.

After there is a vocabulary built or merged into your program (be sure you have it saved on disk) you will of course want to increase it. Type in another 830 foreign words from another sample of text, save them, and go through options 11 and 12 once again. Now select Option 13, UNDEFINED WORDS. The computer will list the words in reverse alphabetical order (z to a) starting with the most frequent and ending with those that occur only once. You will find this feature extremely convenient, since this is also the order in which dictionaries are arranged (that is, if you start from the back of the book). These can then be printed out, defined, and added to the DATA statements.

In selecting substitute words for our DATA file, we should try to select definitions that are most likely to relate to that subject, and ignore other meanings. Of course, you are likely to encounter some words with multiple meanings that could go either way. In such events try to find a word that can, to your mind at least, mean roughly the same thing. That failing, give

(See Page 9)

TRANSLATOR—

(Continued from Page 8)

both meanings, separated by a slash, and postpone the decision until after you see how it nestles among the other words of the translation.

How easy will it be to understand what your computer has translated? It should be obvious that great English literature won't be the result of your first attempt...not likely your last attempt either. But you should have an easier time comprehending what has been translated, at least to the same extent as if I had translated Laussedat's letters using the French-English dictionary. Of course, some phrases will make no sense at all. An example could be the French phrase "nous avions" ("we were having"), which your computer will likely translate as "us airplanes". Unraveling the meaning of such phrases is beyond the capabilities of this program, requiring the teamwork of a human brain with your TI brain.

What is a translation? Here's my guess: it is an attempt to express in the language you understand best what the author would have said if he had been writing or speaking in to you in that language. The concern here is not only with the words he might have chosen to use, but also the ideas that were flowing through his mind — perhaps also the ideas that he perceived were passing through the minds of his audience at the time. It will be seen that this is something beyond the capabilities of even the largest computer in the world because we humans are so illogical and erratic in our way of thinking that only we can really understand each other — and that only to varying degrees at best.

After you have let the computer try its hand at translating some text, you will see how the substituted words associate with one another, some adjustments in word definitions can then be made, and through study of the way the untranslated words relate with the translated words you can also get some clues on the meaning of foreign words that were not in the dictionary as well.

As we begin to understand the way people in other cultures think, the distances separating the peoples of the world will become smaller, and it can and should begin right here in our own TI-99/4A community.

TRANSLATOR

```
10 !******************* !119
20 !*Foreign Language* !229
30 !* TRANSLATOR * !161
40 !******************* !119
50 !!131
60 !Steve Richardson !183
70 !Magna, Utah !069
80 !!131
90 !11Mar95 !115
100 !!131
110 !start foreign words !01
0
120 !with a-z @[\]^_` !248
130 !!131
140 !~~~~~~~~~~~~~~~~~~~ !095
150 W=831 !124
160 DIM B$(831),A$(831)!176
170 !~~~~~~~~~~~~~~~~~~~ !095
180 !!131
190 U$="LAST CHOICE=" :: D$=
" " :: T$=" " :: S$=D$&T$
:: CALL LOAD(-31806,16):: C
ALL LOAD(-31878,0)!211
200 CALL CHAR(130,"000010107
C101000"):: READ M$,L$ :: CA
LL PEEK(-31888,K,S):: IF K=5
9 AND S=227 THEN 250 !176
210 CALL CLEAR :: PRINT "You
must first execute:": : : :
: : : : : : : :"CALL FILES(
1)": :"RUN": : :: END !169
220 D$,C$,Z,J,L,E$,B,F$,G,Z$
,S,K,X,Y,A,W$,C,N,I !020
230 CALL SAY :: CALL COLOR :
: CALL SCREEN :: CALL CLEAR
:: CALL SOUND :: CALL KEY ::
CALL HCHAR :: CALL VCHAR !1
86
240 DATA ===================
=========PRESS <ENTER> TO DO
IT AGAIN OR ANY KEY AND <EN
TER> TO RETURN TO MENU !18
6
250 !@P- !064
260 !!131
270 CALL PEEK(-31974,A,B)::
PRINT : :" ";A*256+B-2487;"V
DP BYTES FREE" :: CALL SOUND
(100,110,0,350,0,500,0)!072
280 !!131
290 !*Special Graphics* !233
300 CALL CHAR(96,"0010080038
447C44"):: CALL CHAR(64,"001
0280038447C44"):: CALL CHAR(
35,"0010087C4078407C"):: CAL
L CHAR(36,"0008107C4078407C"
)!118
310 CALL CHAR(37,"0010287C40
78407C"):: CALL CHAR(94,"001
0280010101010"):: CALL CHAR(
38,"001028007C44447C"):: CAL
L CHAR(42,"0010280044444438"
)!048
320 CALL CHAR(41,"00003C4040
3C0830"):: CALL CHAR(40,"001
824682422226C"):: CALL CHAR(
43,"2054084464544C44")!220
330 !!131
340 ON ERROR 340 :: CALL CLE
AR :: CALL SCREEN(12):: FOR
I=3 TO 8 :: CALL COLOR(I,7,1
2):: NEXT I :: C=INT(28-LEN(
L$))/2 !246
350 DISPLAY AT(2,C):L$: :TAB
(5);"LANGUAGE TRANSLATOR": :
TAB(7);"1 CREATE FILE":TAB(7
);"2 ADD TO FILE":TAB(7);"3
LOAD FILE":TAB(7);"4 TEXT" !
213
360 DISPLAY AT(10,7):"5 LIST
":TAB(7);"6 FIND STRING":TAB
(7);"7 EDIT":TAB(7);"8 SAVE
FILE":TAB(7);"9 DEFINE BLOCK
":TAB(6);"10 TRANSLATE":TAB(
6);"11 ~ALPHABETIZE" !133
370 DISPLAY AT(17,6):"12 ~V
OCABULARY COUNT":TAB(6);"13
~UNDEFINED WORDS":TAB(6);"1
4 FOREIGN WORD LIST":TAB(6);
"15 BREAK": :U$;Z !180
380 DISPLAY AT(23,21):Y-X+1;
"WDS":TAB(2);"SELECT ONE:" :
: ACCEPT AT(24,14):Z :: IF Z
>15 OR Z<1 THEN 340 :: IF Z=
7 THEN 1550 :: DISPLAY AT(24
,1):"IS";Z;"CORRECT? <Y/N>"
!002
390 ACCEPT AT(24,22):W$ :: C
ALL VCHAR(1,1,32,768):: IF W
$<>"y" AND W$<>"Y" THEN 340
!160
400 ON Z GOTO 500,470,1290,5
60,780,1980,1550,1510,430,21
30,1840,1680,890,1060,2070 !
131
```

(See Page 10)

TRANSLATOR—

(Continued from Page 9)

```
410 !!131
420 !*Define Block* !045
430 PRINT "WORD COUNT:",STR
$(1);"-";STR$(N): : :SEG$(U$
,1,11);":",STR$(X);"-";STR$(
Y): : :!037
440 PRINT "YOUR";SEG$(U$,5,7
);":",:: INPUT "":E$ :: A=PO
S(E$,"-",1):: IF A=0 THEN 44
0 :: X=VAL(SEG$(E$,1,A-1))::
Y=VAL(SEG$(E$,A+1,LEN(E$)))
:: GOTO 340 !078
450 !!131
460 !*Add to File* !131
470 N=N+1 :: IF N<9 THEN 510
:: FOR C=N-8 TO N-1 :: PRIN
T TAB(11-LEN(STR$(C)));C,A$(
C):: NEXT C :: GOTO 510 !169
480 !!131
490 !*Create File* !203
500 N=1 :: INPUT "SOURCE: ":
A$(0):: CALL HCHAR(22,1,32,6
4)!247
510 DISPLAY AT(10,7):SEG$(M$
,1,15):TAB(8);"ALPHA LOCK UP
": :TAB(9);"'xx' TO END":TAB
(7);SEG$(M$,1,15)!094
520 ON ERROR 520 :: FOR N=N
TO W :: PRINT TAB(11-LEN(STR
$(N)));N,:: INPUT "":A$(N)::
IF N=W OR A$(N)="xx" THEN 5
30 :: NEXT N !226
530 N=N-1 :: X=1 :: Y=N :: G
OTO 270 !111
540 !!131
550 !*Text* !124
560 PRINT "ON SCREEN:": :TAB
(5);"1 "&L$:TAB(5);"2 ENGLIS
H": : : :" ON PRINTER:": :T
AB(7);"3 "&L$:TAB(7);"4 ENGL
ISH": : : : : :!243
570 INPUT "SELECT ONE: ":C :
: CALL CLEAR :: PRINT A$(0):
: : :: IF C<1 OR C>4 THEN 5
60 !059
580 ON C GOTO 590,620,680,69
0 !189
590 FOR I=X TO Y :: CALL KEY
(3,K,S):: IF S=0 THEN 610 ::
IF K=81 THEN 670 !125
600 CALL KEY(0,K,S):: IF S=1
THEN 610 ELSE 600 !077
610 PRINT A$(I);" ";:: NEXT
I :: GOTO 670 !217
620 FOR I=X TO Y :: CALL KEY
(3,K,S):: IF S=0 THEN 640 ::
IF K=81 THEN 670 !155
630 CALL KEY(5,K,S):: IF S=1
THEN 640 ELSE 630 !142
640 IF B$(I)<>"" THEN 650 ::
PRINT "<"&A$(I)&"> ";:: GOT
O 660 !218
650 PRINT B$(I)&" ";!012
660 NEXT I !223
670 PRINT : : :: GOTO 720 !1
69
680 OPEN #1:"PIO" :: FOR I=X
TO Y :: PRINT #1:A$(I)&" ";
:: NEXT I :: PRINT #1:"" ::
CLOSE #1 :: GOTO 720 !091
690 OPEN #1:"PIO" :: FOR I=X
TO Y :: IF B$(I)="" THEN 70
0 :: PRINT #1:B$(I)&" ";:: G
OTO 710 !037
700 PRINT #1:"<"&A$(I)&"> ";
!178
710 NEXT I :: PRINT #1:"" ::
CLOSE #1 !137
720 CALL KEY(5,K,S):: PRINT
M$;" <P>=PURGE";:: INPUT "
":W$ :: CALL HCHAR(20,1,32,1
28)!018
730 IF W$<>"P" AND W$<>"p" T
HEN 740 :: FOR I=X TO Y :: B
$(I)="" :: NEXT I !221
740 IF W$="" THEN 560 ELSE 2
70 !136
750 !!131
760 !*List* !115
770 CALL SOUND(99,150,0,520,
0)!029
780 PRINT A$(0): : : :: FOR
I=X TO Y :: CALL KEY(3,K,S):
: IF S=0 THEN 800 :: IF K=81
THEN 820 !219
790 CALL KEY(5,K,S):: IF S=1
THEN 800 ELSE 790 !208
800 PRINT TAB(4-LEN(STR$(I))
);STR$(I);" ";A$(I),B$(I)::
IF A$(I)<>A$(I+1)THEN 810 ::
CALL SOUND(99,770,0,110,0)!
212
810 NEXT I !223
820 CALL KEY(5,K,S):: PRINT
:SEG$(M$,1,84):: IF Z<>14 TH
EN 830 :: PRINT " EXIT";:: G
OTO 840 !187
830 PRINT SEG$(M$,85,17);!21
3
840 INPUT "":W$ :: CALL HCHA
R(20,1,32,128)!028
850 IF W$="" THEN 780 :: IF
Z=14 THEN 1240 :: GOTO 270 !
142
860 !!131
870 !*Undefined Words* !152
880 PRINT TAB(9);"PLEASE WAI
T" :: FOR I=1 TO N :: A$(I)=
STR$(VAL(B$(I))*10)&" "&A$(I
):: NEXT I :: GOTO 1000 !046
890 PRINT TAB(9);"PLEASE WAI
T": : : : : : : : :: L=1 ::
FOR I=1 TO N :: A=ASC(A$(I))
:: IF A<64 OR A>122 THEN 940
:: A=A AND 95 !239
900 ON A-63 GOSUB 2200,2210,
2220,2230,2240,2250,2260,227
0,2280,2290,2300,2310,2320,2
330,2340,2350,2360,2370,2380
,2390,2400,2410,2420,2430,24
40,2450,2460,2470,2480,2490,
2500,2510 !154
910 CALL KEY(0,K,S):: IF S=0
THEN 920 :: INPUT T$:W$ ::
IF W$<>"" THEN 270 !234
920 READ F$,E$ :: IF F$="99"
THEN 930 :: W$=F$&"." :: IF
A$(I)<>F$ AND A$(I)<>W$ THEN
920 :: CALL SOUND(10,125,0)
:: GOTO 940 !122
930 CALL SOUND(10,450,0):: A
$(L)=STR$(VAL(B$(I))*10)&"
"&A$(I):: PRINT A$(L):: L=L+
1 !168
940 NEXT I :: G=L-1 :: PRINT
: :!246
950 PRINT SEG$(M$,1,82):" QU
IT";:: INPUT "":W$ :: IF W$<
>"" THEN 1000 !180
960 CALL CLEAR :: FOR I=1 TO
G :: PRINT A$(I)!017
970 CALL KEY(0,K,S):: IF S=0
THEN 990 !037
980 CALL KEY(0,K,S):: IF S=1
THEN 990 ELSE 980 !071
990 NEXT I :: PRINT : : :: G
OTO 950 !241
1000 PRINT : : :: INPUT "WAN
T A PRINTOUT? <Y/N> ":Z$ ::
IF Z$<>"Y" AND Z$<>"y" THEN
1260 !192
1010 K=INT(G/2):: OPEN #1:"P
IO" :: PRINT A$(0): : :: PRI
```

(See Page 11)

TRANSLATOR—

(Continued from Page 10)

```
NT #1:A$(0): : :: FOR L=1 TO
K :: PRINT A$(L),A$(L+K)::
PRINT #1:A$(L);TAB(40);A$(L+
K):: NEXT L :: IF K=G/2 THEN
1030 !080
1020 PRINT ,A$(G):: PRINT #1
:TAB(40);A$(G)!150
1030 PRINT #1:"" :: CLOSE #1
:: GOTO 1260 !233
1040 !!131
1050 !*Foreign Word List* !0
25
1060 IF F$="" THEN 1070 :: I
F F$<>"99" THEN 1080 !253
1070 IF A$(1)<>"" THEN 1170
:: B=INT(W/2.5)!222
1080 PRINT S$;"1 WORD LIST":
S$;"2 DICTIONARY": : : : ::
INPUT S$:G :: CALL CLEAR ::
IF G=1 THEN 1130 :: GOSUB 11
20 :: FOR N=1 TO B !084
1090 READ F$,E$ :: IF F$="9"
THEN 1090 :: IF F$="99" THE
N 1110 :: A$(N)=F$&" "&E$ !2
10
1100 NEXT N !228
1110 X=1 :: B,Y,N,I=N-1 :: G
OSUB 1870 :: FOR I=1 TO N ::
A=POS(A$(I)," ",1):: B$(I)=
SEG$(A$(I),A+1,LEN(A$(I)))::
A$(I)=SEG$(A$(I),1,A-1):: N
EXT I :: GOTO 770 !206
1120 PRINT TAB(9);"PLEASE WA
IT" :: FOR N=1 TO B :: A$(N)
="" :: B$(N)="" :: NEXT N ::
PRINT TAB(11);"LOADING" ::
RETURN !149
1130 B=W :: GOSUB 1120 :: FO
R N=1 TO B !148
1140 READ F$,E$ :: IF F$="9"
THEN 1140 :: IF F$="99" THE
N 1160 :: A$(N)=F$ !117
1150 NEXT N !228
1160 X=1 :: B,Y,N,I=N-1 :: G
OSUB 1870 :: GOTO 770 !162
1170 RESTORE 2420 :: READ L$
:: CALL CLEAR :: C=0 !050
1180 READ F$,E$ :: IF F$="9"
THEN 1180 :: IF F$="99" THE
N 1220 :: CALL KEY(0,K,S)::
IF S=0 THEN 1200 !137
1190 CALL KEY(0,K,S):: IF S=
1 THEN 1200 ELSE 1190 !237
1200 C=C+1 :: IF ASC(F$)=A T
HEN 1210 :: CALL SOUND(90,29
0,0,190,0,770,0):: PRINT SEG
$(M$,1,28)!188
1210 PRINT F$,E$ :: :: A=ASC
(F$):: GOTO 1180 !073
1220 PRINT :S$;C;"WORDS": :!
056
1230 PRINT SEG$(M$,1,84):" E
XIT";:: INPUT "":W$ :: IF W$
="" THEN 1170 :: IF F$="99"
THEN 270 ELSE 1260 !194
1240 INPUT "WANT A PRINTOUT?
<Y/N> ":W$ :: IF W$<>"Y" AN
D W$<>"y" THEN 1260 !047
1250 OPEN #1:"PIO",OUTPUT ::
FOR I=X TO Y :: PRINT #1:I,
A$(I);TAB(30);B$(I):: NEXT I
:: CLOSE #1 !147
1260 STOP !152
1270 !!131
1280 ! *Load File* !023
1290 CALL KEY(3,K,S):: PRINT
"{LOAD FILE}": : :: DISPLAY
AT(2,1):"PRECEDE FILENAME W
ITH '=' IF":"YOU ARE LOADING
A TEXT FILE." !031
1300 ON ERROR 1340 :: INPUT
"DEVICE NAME:":D$ :: J=ASC(D
$):: IF J<>61 THEN 1310 :: D
$=SEG$(D$,2,LEN(D$))!176
1310 N=1 :: CALL KEY(5,K,S):
: IF POS(D$,"DSK",1)>0 THEN
1320 :: D$="DSK1."&D$ !060
1320 IF J=61 THEN 1360 !139
1330 OPEN #1:D$,INPUT :: FOR
N=0 TO W :: LINPUT #1:A$(N)
:: IF EOF(1)THEN 1340 :: NEX
T N !098
1340 IF J=61 THEN N=N-1 !131
1350 A$(N+1)="xx" :: X=1 ::
Y=N :: CLOSE #1 :: CALL KEY(
5,K,S):: GOTO 270 !006
1360 OPEN #1:D$,INPUT :: PRI
NT :: LINPUT "SOURCE: ":A$(0
):: PRINT !218
1370 LINPUT #1:E$ :: CALL KE
Y(0,K,S):: IF S OR EOF(1)THE
N 1340 !250
1380 IF LEN(E$)=0 OR ASC(E$)
=46 OR POS(E$," .",1)THEN 13
70 !252
1390 E$=E$&" " :: FOR C=1 TO
LEN(E$):: A=ASC(SEG$(E$,C,1
)):: IF A>47 AND A<58 OR A>6
3 AND A<123 THEN A=A OR 32 !
193
1400 IF A=33 OR A=59 OR A=63
THEN A=46 :: GOTO 1450 !243
1410 IF A=44 OR A=45 OR A=58
THEN A=32 :: GOTO 1450 !239
1420 IF A=42 OR A=46 OR A=39
THEN 1460 !193
1430 IF A>122 OR A=13 OR A=4
4 OR A=45 OR A=34 THEN 1470
!201
1440 IF A<>32 THEN 1460 !165
1450 GOSUB 1480 :: GOTO 1470
!177
1460 A$(N)=A$(N)&CHR$(A):: I
F A=46 THEN GOSUB 1480 !064
1470 NEXT C :: GOTO 1370 !00
9
1480 IF LEN(A$(N))=0 THEN 14
90 :: PRINT USING "##### "&A
$(N):N :: N=N+1 :: IF N=W TH
EN 1340 !041
1490 RETURN !136
1500 !*Save File* !006
1510 CALL KEY(3,K,S):: PRINT
"{SAVE FILE}": : :: INPUT "
DEVICE NAME: ":D$ :: IF POS(
D$,"DSK",1)=0 THEN D$="DSK1.
"&D$ !073
1520 OPEN #1:D$,OUTPUT :: FO
R I=0 TO N :: PRINT #1:A$(I)
:: NEXT I :: CLOSE #1 :: CAL
L KEY(5,K,S):: GOTO 270 !235
1530 !!131
1540 !*Edit* !093
1550 CALL CLEAR !209
1560 PRINT "{EDIT}": :" S=
SHOW":" I=INSERT":" D=DE
LETE":" E=EDIT": : : :: PR
INT "COMBINE FUNCTION";CHR$(
130);"WORD NUMBER": : :: INP
UT " D372, FOR EXAMPLE: ":W
$ !215
1570 CALL HCHAR(16,1,32,256)
:: IF LEN(W$)<2 THEN 1560 ::
A=VAL(SEG$(W$,2,LEN(W$)))::
W$=CHR$(ASC(W$)AND 95)!135
1580 IF W$="S" THEN 1630 ::
IF W$="I" THEN 1600 :: IF W$
="D" THEN 1610 :: IF W$="E"
THEN 1620 !247
1590 CALL CLEAR :: GOTO 1560
!192
1600 FOR I=N TO A STEP -1 ::
A$(I+1)=A$(I):: NEXT I :: A
```

(See Page 12)

TRANSLATOR—

(Continued from Page 11)

```
$(A)="" :: IF N=W-1 THEN 163
0 :: N,Y=N+1 :: GOTO 1630 !1
99
1610 FOR I=A TO N+1 :: A$(I)
=A$(I+1):: NEXT I :: A$(N+1)
="" :: N,Y=N-1 :: GOTO 1630
!219
1620 PRINT :A;A$(A): : :A;::
INPUT "":A$(A)!016
1630 IF A>2 THEN 1640 :: A=3
!228
1640 PRINT : : : :: FOR I=A-
3 TO A+3 :: PRINT TAB(5);I;A
$(I):: NEXT I :: PRINT : :!1
84
1650 INPUT M$:W$ :: CALL HCH
AR(20,1,32,128):: IF W$="" T
HEN 1560 ELSE 270 !188
1660 !!131
1670 !*vocabulary count* !08
8
1680 PRINT T$;"ORDER DESIRED
:": : :S$;"1 BY ALPHABET":S$
;"2 BY FREQUENCY": : : :T$;"
SELECT ONE:";!212
1690 CALL HCHAR(24,19,32)::
CALL KEY(0,K,S):: CALL HCHAR
(24,19,50):: IF S=0 THEN 169
0 !084
1700 IF K<>13 AND K<>49 AND
K<>50 THEN 1690 !005
1710 IF K<>13 THEN 1720 :: K
=50 !109
1720 IF K<49 OR K>50 THEN 17
00 :: CALL CLEAR :: J=K-48 !
252
1730 PRINT TAB(9);"PLEASE WA
IT": : : : : : : : : : : : :
: B,C=1 :: FOR I=1 TO N :: I
F POS(A$(I+1),".",1)=0 THEN
1740 :: A$(I+1)=SEG$(A$(I+1)
,1,LEN(A$(I+1))-1)!119
1740 IF A$(I)<>A$(I+1)THEN 1
750 :: C=C+1 :: GOTO 1760 !1
29
1750 B$(B)=STR$(C/10)&" "&A$
(I):: A$(I)="" :: B=B+1 :: C
=1 :: DISPLAY AT(24,1):N-I !
074
1760 NEXT I :: FOR I=1 TO N
:: A$(I)=B$(I):: B$(I)="" ::
NEXT I :: IF J=1 THEN 1770
:: A=9 :: GOSUB 1860 !249
1770 PRINT TAB(10);"COUNTING
": : : : : : : : : : : : : ::
FOR I=1 TO N :: DISPLAY AT(2
4,1):N-I :: A=LEN(A$(I))+1 :
: C=POS(A$(I)," ",1)+1 !144
1780 B$(I)=SEG$(A$(I),1,C-1)
:: A$(I)=SEG$(A$(I),C,A-C)::
NEXT I :: CALL SOUND(10,430
,0):: Y,N=B-1 !010
1790 FOR I=1 TO N :: CALL KE
Y(0,K,S):: IF S=0 THEN 1810
!107
1800 CALL KEY(0,K,S):: IF S=
1 THEN 1810 ELSE 1800 !183
1810 PRINT VAL(B$(I))*10;TAB
(5);A$(I):: NEXT I :: PRINT
:S$;N;"WORDS INTRODUCED": :
:: INPUT M$:W$ :: G=N :: CAL
L HCHAR(20,1,32,128):: IF W$
="" THEN 1790 :: IF L$="FORE
IGN" THEN 880 :: GOTO 270 !1
45
1820 !!131
1830 !*alphabetize* !096
1840 A=0 :: GOSUB 1860 :: GO
TO 270 !245
1850 !(Shell Sort) !148
1860 X=1 :: Y,I=N !243
1870 PRINT TAB(11);"SORTING"
: : : : : : : : : : : : :: G
=I*1.5 !071
1880 G=INT(G/2):: DISPLAY AT
(24,1):G :: IF G=0 THEN 1940
!185
1890 FOR C=1 TO I-G :: J=C :
: K=J+G !052
1900 K=J+G :: IF A<>9 THEN 1
910 :: IF A$(J)>=A$(K)THEN 1
930 :: GOTO 1920 !056
1910 IF A$(J)<=A$(K)THEN 193
0 !068
1920 C$=A$(J):: A$(J)=A$(K):
: A$(K)=C$ :: J=J-G :: IF J>
0 THEN 1900 !080
1930 NEXT C :: GOTO 1880 !00
9
1940 C$="" !236
1950 RETURN !136
1960 !!131
1970 !*Find String* !239
1980 INPUT "STRING: ":W$ ::
PRINT : :TAB(10);"NO.","WORD
": : :: FOR I=X TO Y :: A=PO
S(A$(I),W$,1)!127
1990 IF A=0 THEN 2000 :: PRI
NT TAB(13-LEN(STR$(I)));STR$
(I),A$(I):: CALL SAY(STR$(I)
)!254
2000 NEXT I !223
2010 PRINT : :!006
2020 INPUT M$:W$ !051
2030 CALL HCHAR(20,1,32,128)
!017
2040 IF W$="" THEN 1980 ELSE
270 !025
2050 !!131
2060 !*Break* !188
2070 DISPLAY AT(10,2):"TYPE
'SIZE' FOR BYTES FREE" :: FO
R C=1 TO 50 :: NEXT C :: DIS
PLAY AT(12,4):"TYPE 'CON' TO
CONTINUE" !085
2080 FOR C=1 TO 100 :: NEXT
C !239
2090 BREAK !142
2100 GOTO 270 !094
2110 !!131
2120 !*translate* !165
2130 PRINT TAB(9);"TRANSLATI
NG": : : : : : : : : : : : :
: FOR I=X TO Y :: PRINT STR$
(I);A$(I),:: A=ASC(A$(I)):
IF A<64 OR A>122 THEN 2170 :
: A=A AND 95 !226
2140 ON A-63 GOSUB 2200,2210
,2220,2230,2240,2250,2260,22
70,2280,2290,2300,2310,2320,
2330,2340,2350,2360,2370,238
0,2390,2400,2410,2420,2430,2
440,2450,2460,2470,2480,2490
,2500,2510 !154
2150 IF POS(A$(I),".",1)>0 T
HEN 2190 !003
2160 READ F$,E$ :: IF F$="9"
THEN 2170 :: IF F$<>A$(I)TH
EN 2160 :: B$(I)=E$ :: PRINT
B$(I):: GOTO 2180 !158
2170 PRINT "<";A$(I);">" ::
CALL KEY(0,K,S):: IF S=1 THE
N 1850 !049
2180 NEXT I :: GOTO 270 !191
2190 READ F$,E$ :: IF F$="9"
THEN 2170 :: F$=F$&"." :: I
F F$<>A$(I)THEN 2190 :: B$(I
)=E$&"." :: PRINT B$(I):: GO
TO 2180 !044
2200 RESTORE 6400 :: RETURN
!128
2210 RESTORE 6500 :: RETURN
!228
```

(See Page 13)

TRANSLATOR—

(Continued from Page 12)

```
2220 RESTORE 6600 :: RETURN
!072
2230 RESTORE 6700 :: RETURN
!173
2240 RESTORE 6800 :: RETURN
!017
2250 RESTORE 6900 :: RETURN
!117
2260 RESTORE 7000 :: RETURN
!218
2270 RESTORE 7100 :: RETURN
!062
2280 RESTORE 7200 :: RETURN
!163
2290 RESTORE 7300 :: RETURN
!007
2300 RESTORE 7400 :: RETURN
!107
2310 RESTORE 7500 :: RETURN
!208
2320 RESTORE 7600 :: RETURN
!052
2330 RESTORE 7700 :: RETURN
!153
2340 RESTORE 7800 :: RETURN
!153
2350 RESTORE 7900 :: RETURN
!097
2360 RESTORE 8000 :: RETURN
!198
2370 RESTORE 8100 :: RETURN
!042
2380 RESTORE 8200 :: RETURN
!143
2390 RESTORE 8300 :: RETURN
!243
2400 RESTORE 8400 :: RETURN
!087
2410 RESTORE 8500 :: RETURN
!188
2420 RESTORE 8600 :: RETURN
!032
2430 RESTORE 8700 :: RETURN
!132
2440 RESTORE 8800 :: RETURN
!233
2450 RESTORE 8900 :: RETURN
!077
2460 RESTORE 9000 :: RETURN
!178
2470 RESTORE 9100 :: RETURN
!022
2480 RESTORE 9200 :: RETURN
!122
2490 RESTORE 9300 :: RETURN
!223
2500 RESTORE 9400 :: RETURN
!067
2510 RESTORE 9500 :: RETURN
!168
5000 DATA FOREIGN !108
```

CREATE DATA

```
10 !********************** !
031
20 !* Create  DATA  for  * !
108
30 !* TRANSLATOR Program * !
025
40 !********************** !
031
50 !!131
60 !Steve Richardson !183
70 !Magna, Utah !069
80 !!131
90 !11Mar95 !115
100 !!131
110 V=0 !v=1 to insert DATA
     with odd line NUMbers.
 !011
120 I=2 !=line NUM interval
!255
130 !!131
140 !Start foreign wds with
!229
150 !  a-z,[,\,],^,_,`,or @.
 !159
160 !!131
170 GOTO 200 :: A$,B$,D$,E$,
F$,L$,N$,S$,W$ :: A,B,C,E,I,
K,P,S,V :: CALL CHAR :: CALL
 CLEAR :: CALL KEY :: CALL L
OAD :: CALL SAY :: CALL SOUN
D :: CALL VCHAR :: DIM U(31)
:: !@P- !097
180 !!131
190 !*Special Graphics* !233
200 CALL CHAR(96,"0010080038
447C44"):: CALL CHAR(64,"001
0280038447C44"):: CALL CHAR(
35,"0010087C4078407C"):: CAL
L CHAR(36,"0008107C4078407C"
)!118
210 CALL CHAR(37,"0010287C40
78407C"):: CALL CHAR(94,"001
0280010101010"):: CALL CHAR(
38,"001028007C44447C"):: CAL
L CHAR(42,"0010280044444438"
)!048
220 CALL CHAR(41,"00003C4040
3C0830"):: CALL CHAR(40,"001
824682422226C"):: CALL CHAR(
43,"2054084464544C44")!220
230 !!131
240 CALL LOAD(-31806,16)!
    (Disables QUIT key) !018
250 CALL LOAD(-31878,0)!
    (Disables Sprites.) !111
260 CALL KEY(3,K,S):: S$=CHR
$(179)&CHR$(200)!124
270 CALL CLEAR :: DISPLAY AT
(9,2):"Create Translator Dat
abase": : : :: INPUT "Filena
me: ":N$ :: PRINT :: IF POS(
N$,"DSK",1)THEN 280 :: N$="D
SK1."&N$ !116
280 PRINT "Will you MERGE th
is into" :: INPUT " another
Datafile? <Y/N>: ":A$ :: A=A
SC(A$)!100
290 CALL KEY(5,K,S):: PRINT
:: INPUT "What Language? ":L
$ :: GOSUB 570 :: IF A<>89 T
HEN 320 :: PRINT TAB(9);"Ple
ase Wait" !011
300 OPEN #1:W$,INPUT :: INPU
T #1:A$ :: FOR A=0 TO 31 ::
INPUT #1:U(A):: NEXT A :: CL
OSE #1 :: B$=SEG$(N$,6,LEN(N
$)):: IF B$<>A$ THEN 320 ::
CALL CLEAR !029
310 PRINT "Previous filename
=";A$: :"Current  filename="
;B$: : : : : :"You shouldn't
 use the same":" name as you
r previous":" file.": :"Plea
se try again.": : : :: END !
007
320 OPEN #1:N$,VARIABLE 163,
OUTPUT :: PRINT #1:CHR$(19)&
CHR$(136)&CHR$(147)&CHR$(200
)&CHR$(LEN(L$))&L$&CHR$(0)::
 CALL VCHAR(1,1,32,768)!015
330 PRINT "Type only in lowe
rcase;":"Separate words with
 <SPACE>;":"<ENTER> to close
 file.": : : :L$,"English" !
142
340 ON ERROR 340 :: INPUT ""
:W$ :: GOSUB 510 !058
350 DISPLAY AT(23,1):"" :: P
RINT :: K=1 :: A,B=ASC(A$)AN
```

(See Page 14)

TRANSLATOR—

(Continued from Page 13)

```
D 95 :: CALL SAY(CHR$(A))::
P=A*100+U(A-64)+V :: GOTO 38
0 !232
360 ON ERROR 360 :: INPUT ""
:W$ :: GOSUB 510 :: A=ASC(A$
)AND 95 !112
370 IF A<>B THEN 430 !174
380 FOR C=1 TO LEN(B$):: E=A
SC(SEG$(B$,C,1)):: IF E=32 O
R E=39 OR E=47 THEN 400 !035
390 E=E AND 95 :: IF E<32 TH
EN 410 !136
400 E$=E$&CHR$(E)!208
410 NEXT C :: F$=S$&CHR$(LEN
(A$))&A$&S$&CHR$(LEN(E$))&E$
 !175
420 DISPLAY AT(23,1):A$,E$ :
: D$=D$&F$ :: E$="" :: DISPL
AY AT(2,18):" ";CHR$(B);P;ST
R$(K):: IF LEN(D$)<140 THEN
360 ELSE 440 !234
430 IF LEN(D$)<2 THEN 350 !1
92
440 ON ERROR STOP :: PRINT #
1:CHR$(INT(P/256))&CHR$(P-25
6*INT(P/256))&CHR$(147)&SEG$
(D$,2,LEN(D$))&CHR$(0):: D$=
"" :: P=P+I+V :: U(B-64)=P-B
*100 :: IF S=89 THEN 460 !00
8
450 IF A=B THEN 360 ELSE 350
 !080
460 FOR S=6499 TO 9599 STEP
100 :: P=INT(S/256):: PRINT
#1:CHR$(P)&CHR$(S-256*P)&CHR
$(147)&CHR$(200)&CHR$(1)&"9"
&S$&CHR$(1)&"9"&CHR$(0):: NE
XT S !135
470 PRINT TAB(9);"Please Wai
t" :: PRINT #1:CHR$(39)&CHR$
(15)&CHR$(147)&CHR$(200)&CHR
$(2)&"99"&S$&CHR$(2)&"99"&CH
R$(0):CHR$(255)&CHR$(255)!12
2
480 CLOSE #1 :: GOSUB 570 ::
 OPEN #1:W$,OUTPUT :: PRINT
#1:SEG$(N$,6,LEN(N$)):: FOR
S=64 TO 95 :: PRINT #1:U(S-6
4):: NEXT S :: CLOSE #1 !198
490 PRINT : : : :"Your file
was saved as":" ";N$;".": :"
You can MERGE other files":"
 in ";L$;" to build a":" lan
guage file to MERGE":" with
";SEG$(N$,1,5)&"TRANSLATOR."
 !225
500 PRINT :"(The file ";W$:"
 keeps track of locations":"
 for additional DATA.)": : :
 : : : :: END !107
510 IF LEN(W$)THEN 530 !206
520 INPUT "Do you really wan
t out? ":W$ :: IF LEN(W$)=0
THEN 520 :: S=ASC(W$)AND 95
:: IF S<>89 THEN 360 ELSE 44
0 !106
530 S=POS(W$," ",1):: IF S=0
 THEN DISPLAY AT(23,1):"
   <discarded>" :: CALL SOU
ND(99,150,0):: GOTO 360 !121
540 A$=SEG$(W$,1,S-1):: B$=S
EG$(W$,S+1,LEN(W$))!233
550 IF ASC(B$)<>32 THEN 560
:: B$=SEG$(B$,2,LEN(B$)):: G
OTO 550 !223
560 K=K+1 :: RETURN !025
570 B$="" :: FOR S=1 TO LEN(
L$):: B$=B$&CHR$(ASC(SEG$(L$
,S,1))AND 95):: NEXT S :: W$
=SEG$(N$,1,5)&"-"&B$ :: RETU
RN !034
```

The following is a sample of the METROPHOTO D/V80 file that is merged to TRANSLATOR. It is included on the May MICROpendium disk but is too long to include here.

METROPHOTO

```
la
photographie
a
fait
depuis
longtemps
de
grands
progres
aux
etats-unis
et
d'
importantes
applications
scientifiques
et
militaires
y
ont
ete
entreprises
plus
tot
qu'
en
europe
c'est
ainsi
que
pendant
la
guerre
de
la
secession
les
```

The following is a sample of the PHO*FRENCH that is saved in MERGE format prior to being translated. It is included on the May MICROpendium disk but is too long to include here.

PHO*FRENCH

```
5000 DATA French
6400 DATA `,AT/WITH/BY
6499 DATA 9,9
6500 DATA au,HAS,aux,TO,aute
urs,AUTHORS,annee,YEAR,afin,
SO THAT,axe,AXIS,attendre,A
WAIT,avec,WITH,autre,OTHER,
autres,OTHERS
6502 DATA avoir,HAVE,arc,AR
CH,appareil,CAMERA,accompag
ne,ACCOMPANY,aide,HELP,aus
sitot,SOON,apres,AFTER,ains
i,THIS,armee,ARMY
6504 DATA avions,AIRPLANES/
WERE HAVING,aussi,ALSO,auqu
el,TO WHOM,arrivait,ARRIVED
,arpenteur,SURVEYOR,apparei
ls,CAMERAS,apercevoir,PERCE
IVE
6506 DATA abrupts,STEEP,ava
nce,PROGRESS,attente,EXPECT
ATION,agreer,TO ACCEPT,adre
sser,SEND,accueillerez,WELC
OME,autrement,OTHERWISE
6508 DATA anglaise,ENGLISH,
an,YEAR,ailleurs,ELSEWHERE,
agreable,PLEASANT,abandonn
e,DESERTED
6599 DATA 9,9
```

THE ART OF ASSEMBLY — PART 47

CALL FILES from XB

By BRUCE HARRISON

We've talked about "CALL" operations before in this column, but this month we're showing a real working solution to a real problem, not just playing around. Many of us who've used TI Extended BASIC have wished that we could execute a CALL FILES from within a running program. TI did not allow that to be done, and for good reasons, which we'll get to shortly. There are two reasons why one might want to use CALL FILES within a program: First, one might have a program that needs to have more than three files open at some time during its execution. Second, one might have a program that uses so much string space (STACK memory, aka VDP RAM) that it needs to have the number of files set to 1 or 2 to allow more room for its string operations.

TI HAD REASONS

TI Extended BASIC uses VDP RAM for a number of purposes while a program is running. A portion is used for the screen display and character definitions, sprite tables, color tables and such. This stuff is stored in the lower parts of VDP BASIC at addresses below >1000. Another part, in the higher addresses, is used for file operations. That normally extends from >37D8 through >3FFF, which is the end of VDP BASIC. The part in between 11000 and >37D7 is used for string variables, the symbol tables for string and numeric variables, and CALLs. When a program starts running, XB performs a prescan operation, and sets up the symbol tables for variables and calls starting at >37D7 and working downward in VDP memory until all the variables and calls have been assigned places in these tables. This is the case for a "normal" situation where the default number of files (3) is allowed for. If a CALL FILES with an argument other than 3 had been performed before the program started, then the symbol tables would be formed at some address other than >37D7. Here is a partial "mapping" for VDP BASIC in the default case:

How can we trick XB into doing what we want it to do? The solution is another of those things that TI forgot to tell us.

VDP ADDRESSES	CONTENT
0000 - 1000	Screen, characters, sprites, etc.
1000 - xxxx	Reserved for strings, etc.
xxxx - 37D7	Symbol tables
37D7 - 3FFF	Reserved for file operations

Here we've used "xxxx" to indicate the end of the space used for the symbol tables. When string variables are used in the program, their "content" gets placed into the space beginning at xxxx, working downward toward the limit at about >1000.

Now let's suppose that while this program is running we execute a CALL FILES(4). That action would wipe out 518 bytes of VDP before >37D7, thus destroying all or part of the symbol tables that the program needs to find its variables. After that, the program simply could not run. It would stop with SYNTAX ERROR messages when it failed to find a variable in the symbol table.

THE DILEMMA

There, in a nutshell, is the dilemma we face. If we devise an Assembly routine to get around TI's prohibition of CALL FILES

(See Page 16)

Sidebar 47

```
* SIDEBAR 47
* A WAY TO MAKE CALL FILES POSSIBLE
* FROM WITHIN A RUNNING XB PROGRAM
* WITHOUT CAUSING OTHER PROBLEMS
* CALLS FILES (1-9) FROM XB
* 21 SEP 94
* by Bruce Harrison
* Public Domain
* USE VIA CALL LINK("CALLF",NF,CL)
* NF AND CL CAN BE NUMBERS OR VARIABLES
* NF MUST EVALUATE BETWEEN 1 AND 9, INCLUSIVE
* CL MUST BE A LINE NUMBER HIGHER THAN THE LINE
* CONTAINING THE CALL LINK

   DEF CALLF          DEFINE ENTRY POINT
*
* REQUIRED EQUATES
*
PAB EQU >1000         PERIPHERAL ACCESS BLOCK VDP AD-
DRESS
CALPNT EQU >832C      POINTER FOR CALL FILES
PABPNT EQU >8356      POINTER FOR DSRLNK
GPLWS EQU >83E0       GPL WORKSPACE
NUMREF EQU >200C      NUMERIC REFERENCE
VMBW EQU >2024        VDP MULTI BYTE WRITE
XMLLNK EQU >2018      XML LINKAGE
CFI EQU >12B8         CONVERT FLOATING POINT TO INTEGER
ERR EQU >2034         XB ERROR REPORT
BADVAL EQU >1E00      ERROR CODE FOR "BAD VALUE"
FAC EQU >834A         FLOATING POINT ACCUMULATOR
IOERR EQU >2400       I/O ERROR CODE FOR XB ERROR ROU-
TINE
*
* MAIN CODE SECTION
*
CALLF LWPI WS         LOAD OUR WORKSPACE
   CLR R0   CLEAR R0 FOR NUMREF
   LI R1,1  FIRST PARAMETER
   BLWP @NUMREF       GET THE NUMBER OF FILES
   BLWP @XMLLNK       USE XML LINKAGE
   DATA CFI CONVERT FLOATING POINT TO INTEGER
   MOV @FAC,R8        PUT NUMBER IN R8
   INC R1   SECOND PARAMETER
   BLWP @NUMREF       USE NUMREF
   BLWP @XMLLNK       THEN XML
   DATA CFI CONVERT TO INTEGER
```

THE ART OF ASSEMBLY—

(Continued from Page 15)

within a program, we still wind up with a program that will not run under XB after the CALL LINK to our Assembly routine. How can we trick XB into doing what we want it to do? The solution is another of those things that TI forgot to tell us.

Those familiar with TI Extended BASIC know that one can use the RUN command as a statement in a program. Most commonly this is used with a file name, as in RUN "DSK1.NEWPROG," to load and run another program. TI told us this would work, and it does. TI also told us that we can start the running of a program that's currently in memory at a specific line number, as in RUN 120 from Command mode. What the books didn't tell us is that the RUN command can be used in any of its forms within the program. We found that out only by trying it! We half expected that when a program contained a line like 130 RUN 150, we'd get a SYNTAX ERROR IN 130 message. That didn't happen! After a few carefully controlled experiments, we were able to convince ourselves that such a statement in an XB program does just what we'd hoped it would do. That is, it starts from scratch, performs its prescan operation, and then starts the program at the line number given in the statement. You can prove this for yourself by running the following short test program in XB:

```
100 A=15
110 RUN 130
120 A=25
130 PRINT A
```

When this program is run, it will print the value of A after executing the RUN 130 command, so you'll see 0 printed as the value of A. That's so because the prescan process that happens when line 110 executes sets all numeric variables to zero, thus eliminating the value 15 that was assigned to A in line 100. One can then prove another small point by typing in RUN 120 <Enter> from Command mode. This time the value 25 will get printed, as you'd expect. This simply shows that in the first run, line 120 never got executed at all. Thus the solution to our dilemma was to combine the CALL FILES Assembly routine with a statement that would RUN at a specific line number after the CALL FILES had executed.

Before we go any farther, here's a warning for all those who use other versions of Extended BASIC or third-party disk controllers. We do not have any XBs other than TI Extended BASIC, nor do we have any Myarc or CorComp controllers! All of what we're presenting in this month's column has been tested and worked "as advertised" in TI Extended BASIC, and using a TI Disk Controller. None of it has been tested with Myarc XB, Super Extended BASIC, Myarc or CorComp disk controllers, or any other variant. Of course it might work with those, but please don't complain to us if it doesn't.

THE SIDEBAR

Today's sidebar gives the complete source code for our utility routine CALLF, plus the listing of a demo program that will show that it works. We have distributed a public domain disk called CALLFILES which contains this source file, an object file, demo programs, and complete instructions for using the utility with

(See Page 17)

```
    MOVB @FAC,@FKL1+3 MOVE TO THE "RUN" STRING
    MOVB @FAC+1,@FKL1+4 BOTH BYTES
    LI R9,>8300         POINT AT CPU RAM PAD
    LI R10,SAVPAD AND STORAGE
    LI R4,256           256 BYTES TO MOVE
    BL @MOVBTS          STASH RAM PAD
    CI R8,1  CHECK R8 AGAINST 1
    JLT NUMERR          IF LESS, REPORT ERROR
    CI R8,9  COMPARE TO 9
    JGT NUMERR          IF GREATER, ERROR
    AI R8,>30           ADD >30 MASK
    SWPB R8  SWAP BYTES
    MOVB R8,@PABDT+9 PLACE IN THE TOKENIZED STRING
    LI R0,PAB           POINT AT PAB
    LI R1,PABDT         START OF PAB DATA
    LI R2,FKL1-PABDT INCLUDE THE TOKENS
    BLWP @VMBW          WRITE TO VDP
    MOV R0,@PABPNT MOVE R0 TO >8356
    MOV R0,@CALPNT AND TO >832C
    BLWP @DSRLNK        USE DSR LINK
    DATA >A  WITH DATA FOR A "CALL" SERVICE
    MOV @>8350,R1 CHECK FOR ERROR
    JNE FILERR          IF ERROR, JUMP
    MOV @>8370,R1 ELSE GET NUMBER FROM >8370
    LI R9,SAVPAD POINT AT STORAGE
    LI R10,>8300 AND RAM PAD
    LI R4,256           256 BYTES
    BL @MOVBTS          PUT RAM PAD BACK
    MOV R1,@>8370 PLACE NEW NUMBER IN >8370
    LI R1,FKL1          POINT AT "RUN" TOKENIZED LINE
    MOV R1,@CALPNT PLACE ADDRESS AT >832C
    LWPI >83E0          LOAD GPL WORKSPACE
    B @>006A BRANCH TO GPL INTERPRETER
NUMERR LI R0,BADVAL USE BAD VALUE CODE
    BLWP @ERR           USE XB ERR LINKAGE
FILERR LI R0,IOERR   USE I/O ERROR CODE
    BLWP @ERR           USE XB ERROR REPORT
MOVBTS MOVB *R9+,*R10+ MOVE ONE BYTE
    DEC R4   DEC COUNT
    JNE MOVBTS          IF NOT ZERO., REPEAT
    RT       ELSE RETURN
*
* GENERAL PURPOSE GPL AND DSR LINKS
* BY DOUG WARREN AND CRAIG MILLER
*
GR4 EQU GPLWS+8
GR6 EQU GPLWS+12
STKPNT EQU >8373
LDGADD EQU >60
XTAB27 EQU >200E
GETSTK EQU >166C
*
GPLLNK DATA GLNKWS
    DATA GLINK1
RTNAD DATA XMLRTN
GXMLAD DATA >176C
    DATA >50
GLNKWS EQU $->18
    BSS >08
GLINK1 MOV *R11,@GR4
    MOV *R14+,@GR6
    MOV @XTAB27,R12
    MOV R9,@XTAB27
    LWPI GPLWS
    BL *R4
    MOV @GXMLAD,@>8302(R4)
    INCT @STKPNT
    B @LDGADD
XMLRTN MOV @GETSTK,R4
    BL *R4
    LWPI GLNKWS
    MOV R12,@XTAB27
```

THE ART OF ASSEMBLY—

(Continued from Page 16)

your own XB programs. That's available through the Lima User Group for a nominal fee. Since it's public domain, you or your user group can make copies for anyone who needs it. The source code as shown in the sidebar is designed solely for use as a "CALL LINK" from XB. For those doing pure Assembly programs, number 44 in this series contains all you should need to CALL FILES in those programs.

The source code starts in the normal fashion, by setting the Workspace Pointer to our own workspace. It then clears our R0 in preparation for getting the parameters from the CALL LINK. Both parameters get converted into integers by XMLLNK vector. The first parameter, which is the number of files to be permitted, is kept in R8, while the second, which is a line number, gets placed directly into the two bytes of data at FKL1+3 and FKL1+4

We found through experimentation that the FILES routine itself writes things into areas in RAM Pad, and that this can mess up the return to XB if RAM Pad is not returned to its original state. Therefore, we took the "brute force" approach of simply stashing all 256 bytes of RAM Pad in our own routine's memory space (at SAVPAD) for restoration later. The routine could probably be made to work by saving and restoring only a part of that >8300 area, but we didn't want to put in all the hours required to find out what was the minimum necessary part to save and restore.

Now we take the number of files parameter, still stored in R8, and compare it to the limits 1 and 9. If it's outside those limits, we take a shortcut to an error reporting process, so the user will know there's a bad value in the CALL LINK. We do that by setting our own R0 to >1E00, then simply BLWP to the ERR vector at >2034, which then generates the "BAD VALUE IN xxx" error report for the user.

Given that the parameter is in the correct range, we add the value >30 to it, so that it's correct as a "string" character to represent the number, and place the result (>31 through >39) in our PAB data, replacing the '9' at PABDT+9. Now take a look at the data at label PABDT. You'll notice that this is missing the normal first eight bytes of a PAB data. We found, again by just trying it out, that only the name length, the name, and the parameter part in "tokenized" form are required. You'll note also that the name length is 5, so it represents only the length of the word "FILES", not the length of the entire block. The line in PABDT after the word "FILES" is in XB tokens, so that the FILES routine can use it just as if it were part of an XB command. Thus the opening parenthesis is the byte >B7. This is followed by a >C8 token, meaning that what follows is an unquoted string. The next byte is the length of that string, (always 1 for CALL FILES) and that's followed by the number in the range >31 through >39 that we obtained from the parameter. The last byte is a >B6, which is the token for closing parenthesis.

We now write this whole block, from label PABDT to label FKL1, into VDP BASIC at the PAB location >1000. Now since our R0 already points to the "name length" byte, we place R0 at >8356 for the DSR operation. However, the FILES routine itself

(See Page 18)

```
     RTWP
PUTSTK EQU >50
TYPE EQU >836D
NAMLEN EQU >8356
VWA EQU >8C02
VRD EQU >8800
GR4LB EQU >83E9
GSTAT EQU >837C

DSRLNK DATA DSRWS,DLINK1

DSRWS EQU $
DR3LB EQU $+7
DLINK1 MOV R12,R12
     JNE DLINK3
     LWPI GPLWS
     MOV @PUTSTK,R4
     BL *R4
     LI R4,>11
     MOV R4,@>402(R13)
     JMP DLINK2
     DATA 0
     DATA 0,0,0
DLINK2 MOVB @GR4LB,@>402(R13)
     MOV @GETSTK,R5
     MOVB *R13,@DSRAD1
     INCT @DSRADD
     BL *R5
     LWPI DSRWS
     LI R12,>2000
DLINK3 INC R14
     MOVB *R14+,@TYPE
     MOV @NAMLEN,R3
     AI R3,-8
     BLWP @GPLLNK
DSRADD BYTE >03
DSRAD1 BYTE 00
     MOVB @DR3LB,@VWA
     MOVB R3,@VWA
     SZCB R12,R15
     MOVB @VRD,R3
     SRL R3,5
     MOVB R3,*R13
     JNE SETEQ
     COC @GSTAT,R12
     JNE DSREND
SETEQ SOCB R12,R15
DSREND RTWP
*
* DATA SECTION
*
WS  BSS 32   WORKSPACE
PABDT BYTE 5 CALL NAME LENGTH
     TEXT 'FILES'      NAME
     BYTE >B7,>C8,1,'9',>B6 TOKENIZED CODE FOR (9)
FKL1 BYTE >82,>A9,>C9,0,10,0 TOKENS FOR :: RUN (LINE
NUMBER)
SAVPAD BSS 256        STORAGE FOR CPU RAM PAD
     END

* DEMO PROGRAM TO BE RUN IN XB
* LISTED IN 28 COLUMNS
*
10 CALL INIT
20 CALL LOAD("DSK1.CFXB/O")
30 A=5
40 CALL LINK("CALLF",6,50)
50 PRINT A
60 A=15 :: PRINT A :: CALL P
EEK(-31888,A,B):: PRINT A*25
6+B

```

THE ART OF ASSEMBLY—

(Continued from Page 17)

begins its parsing of the data provided at the location pointed to by 8832C, so we also place our R0 into that pointer before doing the DSRLNK operation. Note that the DSRLNK is followed by a DATA >A, not the DATA 8 used for file operations. This way, DSRLNK will find and execute the FILES routine in the disk controller's ROM.

When we get control back, the new value for the "highest available address in VDP BASIC" will be at >8370, unless there was an error detected, in which case >8350 will be non-zero. If there's no error, we "capture" >8370 into our own R1, restore the contents of RAM Pad, put the new value back into >8370, and we're finished.

This whole thing might not work at all with Myarc or CorComp controllers present, and we'd be especially cautious about trying this on systems that contain hard drives of any kind.

THE TRICKY PART

Now we're almost ready to return control of the computer to Extended BASIC, but if we did that right at this point, our XB program might never be able to find any of its variables. Thus we must trick XB into starting the prescan process over from the top, and then make it restart the program from a line beyond the line containing the CALL LINK("CALLF"...) We do this by a very simple trick given to us by our good friend Harry Wilhelm. Look at the data at label FKL1. That starts with >82, which is the token for a double colon. This will tell XB that what follows is a continuation of the current program line. What follows the >82 is >A9, the token for RUN, followed by the token CC9, which means "the next two bytes are a program line number." Now remember that we placed the second parameter from the CALL I INK into the two bytes just after that >C9. Let's suppose the second parameter was 40. Then the tokens starting at label FKL1 will be equivalent to the XB statement ":: RUN 40". That, of course is what we want XB to do next. We trick XB into doing what's at FKL1 by simply placing the address of FKL1 at >832C, loading the workspace pointer to the GPL Workspace at >83E0, and branching back to the GPL Interpreter at >006A. XB will then dutifully execute a "RUN 40" just as if we'd typed that in Command mode and pressed Enter.

CAUTION

The CALL LINK to CALLF should be done very early in the program, before any values are set into variables. That's important because the CALL LINK to CALLF will result in all variables being zero (for numerics) or null (for strings) after the CALL LINK is completed. In the DEMO XB program that's included in the sidebar, we set a value on the variable A before the CALL LINK just to illustrate that A becomes 0 after the CALL LINK.

The last number printed by the demo program is the value of the word at >8370 after the CALL FILES. This will be changed in increments of 518 for each "file" allowed, as follows:

Files	Number from >8370 (decimal)
1	15331
2	14813
3 (default)	14295
4	13777
...	
9	11187

When a program that uses this CALL LINK ends, the files allowance will remain set to whatever the program did when it began. That condition may or may not be desirable for whatever you do next with XB. You can correct that by including a couple of lines at the end of the program like this:

```
3500 CALL LINK("CALLF",3,3510)
3510 END
```

The line numbers would of course be changed to whatever suits your particular program, but setting the number of files back to 3—which is the default, should allow any other XB program to behave as expected.

Our testing was done on our main TI system, which has the TI disk controller, two floppy drives, two Horizon RAMdisk cards, and a P-GRAM installed. This whole thing might not work at all with Myarc or CorComp controllers present, and we'd be especially cautious about trying this on systems that contain hard drives of any kind. We'd like to hear from any readers who have Myarc or CorComp controllers if they try this out, either through "Reader to Reader" or directly.

For those who want to apply this to existing XB programs, you should know that it will also work correctly in those cases where the original XB program has a "prescan avoidance" feature in it. We were not sure whether that would mess up our "RUN (line number)" statement, but having tested this by adding the prescan avoidance feature (!@P-) to our demo programs from the Public Domain disk, we can assure you it works. (Be sure to include CALL LINK within the prescanned part of the program. The actual CALL LINK to CALLF can be beyond the prescanned part without causing trouble.) If needed, the parameters in the CALL LINK can be given as variables, provided those are assigned suitable values before the CALL LINK is executed.

Once again we've found a way to trick our favorite computer into doing something in spite of itself. We hope this will prove valuable to our XB programmers. Next month's topic is undecided. See you then.

Ticky-Tacky

A Tic-Tac-Toe for team play

By DON STEFFEN

The Philadelphia style Tic-Tac-Toe in the February MICROpendium started me thinking about this Ticky-Tacky-Triky-Treky program. It is a game that can be played by 2-6 players in either two or three teams. It uses a 5X5 grid and has many ways to score points. The outcome is not predictable like Philadelphia-style.

The game uses magnify 2 sprites as the markers, which are changed to the team symbol as play proceeds. I used a little trick to get the appearance of 5 sprites on a line. Players are prompted for their turn and need only touch the key displayed on the grid to mark that spot for their team. When the 25 spots on the board are filled the automatic scoring goes to work to calculate each team's score.

When the scores are displayed you have the option to continue with the game and adding to your score or starting a new game. As each game is continued the starting player is changed to give each team an equal chance at winning.

Scoring possibilities are:

3, 4 or 5 in a row to score 1, 2 or 3 points.

2X2 squares for 2 points or 3X3 triangles for 4 points.

3X4 crosses for 4 points or 4X4 stars or 2X2 circles for 6 points.

4X4 boxes or 4X4 triangles for 7 points.

5X5 triangles or 5X5 boxes for 8 points.

5X5 crosses or 3X3 octagons or 5X4 K patterns for 9 points.

5X3 triangles for 5 points or 3X3 diamonds for 9 points.

The DATA tables at 620-820 are used by the TW, DW and CW subprograms which do the score calculating for each team. The board is checked vertically, horizontally and diagonally for each of the winning combinations, so it takes a few minutes to finish scoring all teams. However, if there are more than 3 teams or 5 players, then the score checking is shortened because the combinations requiring 9 or more spots are not checked. The DATA tables are set up so that the first value read is the point value for that combo and the second value read is the number of spots in that combo and the third value read is the number of starting positions for each combo. Then the next 4 to 12 values are the relation of each spot to the starting spot. Then the last values are the starting positions for each checking loop. If all the spots checked in each loop are that team's marker then the combo value is added to that team's score.

Page Pro 99 was another program I wanted to run off my RAMdisk, and I did the same procedure in another section of my RAMdisk I called DSK8....

There are 20 directions to get 3, 4 or 5 in a row and 181 starting positions for finding the other combos.

The name of the game is just a use of 5 letter words with the tic-tac-toe reference. Maybe I should call it TRICKWARE?

Note: The CW subprogram could also look for and calculate all of the alphabet and number digit patterns that might be in the screen display array. It would need additional data lines to do the 220 possible starting positions, so I left them out of this program.

TICKYTACKY

```
90 ! SAVED AS TICKYTACKY ON
DSTEFEN377 !113
100 ! ***********************
   TICKY-TACKY-TRIKY-TREKY
!104
110 ! BY DON STEFFEN 01MAY95
 !064
120 DIM KP(16),CN(31),TS(6),
A(16),GN(6,3)!093
125 CALL COLOR(3,16,1,4,16,1
,5,16,1,6,16,1,7,16,1,8,16,1
,13,16,1,14,16,1)!023
130 CALL COLOR(9,8,1,10,10,1
,11,11,1,12,12,1):: CALL CLE
AR :: CALL SCREEN(2):: CALL
MAGNIFY(2):: H$=" TICKY-TACK
Y-TRIKY-TREKY" !023
135 FOR X=1 TO LEN(H$):: CAL
L SPRITE(#X,ASC(SEG$(H$,X,1)
),RND*12+3,X*4+20,240,0,-10)
:: CALL SOUND(300,110*2^(X/1
2),0):: NEXT X !218
140 GOSUB 480 :: INPUT "HOW
MANY PLAYERS? 2-6 ":N :: INP
UT "HOW MANY TEAMS? 2-3 ":G
!149
150 CALL CHAR(34,"000000FFFF
",35,"1818181818181818",33,"
181818FFFF181818")! BOARD !1
58
160 RESTORE 540 :: FOR E=1 T
O 16 :: READ KP$ :: CALL CHA
R(E+127,KP$):: NEXT E !240
170 CALL CLEAR :: FOR X=5 TO
 17 STEP 4 :: CALL HCHAR(X,7
,34,19):: NEXT X :: FOR X=10
 TO 22 STEP 4 !103
180 CALL VCHAR(2,X,35,20)::
NEXT X :: FOR X=5 TO 17 STEP
 4 :: FOR Y=10 TO 22 STEP 4
:: CALL HCHAR(X,Y,33):: NEXT
 Y :: NEXT X !254
190 RESTORE 500 :: CALL DELS
PRITE(ALL):: CALL SCREEN(5):
: S=1 :: FOR X=1 TO 5 :: FOR
 Y=1 TO 5 !135
200 READ K :: CALL SPRITE(#S
,K,16,X*32-22,Y*32+20):: CN(
S)=K :: S=S+1 :: NEXT Y !011
210 R=X*4-2 :: READ K1,K2,K3
,K4 :: C=23 :: CALL HCHAR(R,
C,K1):: CALL HCHAR(R+1,C,K2)
:: CALL HCHAR(R,C+1,K3):: CA
```

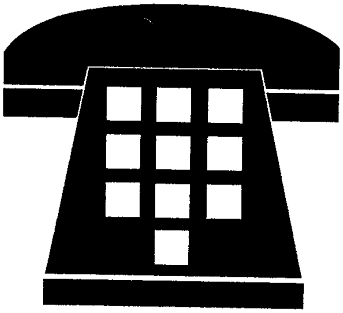
(See Page 20)

TICKY-TACKY—

(Continued from Page 19)

```
LL HCHAR(R+1,C+1,K4)!137
220 NEXT X :: RESTORE 580 ::
 FOR X=1 TO G*4 :: READ K,K$
 :: CALL CHAR(K,K$):: NEXT X
 :: IF NG=1 THEN 250 ELSE CA
LL HCHAR(22,1,32,96)!239
225 ! ~~~~~~~~~~~~~~~~~~~~~
   ASSIGN TEAM & PLAYER PAT
TERNS !199
230 TS(1),TS(2),TS(3)=0 :: F
OR TN=1 TO G :: DISPLAY AT(2
3,1):"TEAM #";TN;" NAME?" ::
 ACCEPT AT(23,20)SIZE(8):N$(
TN):: NEXT TN :: PT=1 !002
240 FOR PN=1 TO N :: DISPLAY
 AT(23,1):"PLAYER #";PN;"NAM
E?" :: ACCEPT AT(23,20)SIZE(
8):PN$(PN):: NEXT PN !038
250 RESTORE 600 :: FOR X=1 T
O N :: CALL CHAR(X*8+80,"FFF
FFFFFFFFFFFF"):: READ NK,K$
 :: CALL CHAR(NK,K$,8*X+89,K
$):: NEXT X !165
260 ! ~~~~~~~~~~~~~~~~~~~~~
  START OF PLAY-FILL SQUARES
 !232
270 FOR PN=1 TO N :: DISPLAY
 AT(23,1):RPT$(CHR$(PN*8+88)
,28):: DISPLAY AT(24,1):PN$(
PN);"'S TURN TO PLAY" !046
280 CALL KEY(3,K,S):: IF S<1
 OR K>80 OR K<49 THEN 280 EL
SE SN=K-48+(K>57)*7 !169285
IF CN(SN)<49 THEN CALL SOUND
(100,110,0):: GOTO 280 !043
290 TN=GN(PN,G):: CALL PATTE
RN(#SN,57+TN):: CALL COLOR(#
SN,TN+7):: CN(SN)=TN+25 :: N
P=NP+1 :: IF SN/5<>INT(SN/5)
THEN 300 !192
295 R=(SN/5)*4-1 :: CALL HCH
AR(R-1,23,TN*8+92):: CALL HC
HAR(R,23,TN*8+93):: CALL HCH
AR(R-1,24,TN*8+94):: CALL HC
HAR(R,24,TN*8+95)!228
300 IF NP>24 THEN PT=PN+1 ::
 GOTO 320 !143
305 NEXT PN :: GOTO 270 !020
310 ! ~~~~~~~~~~~~~~~~~~~~~
   CALCULATE SCORING !092
320 DISPLAY AT(1,1):" BOARD
FULL PRESS ANY KEY" :: CALL
KEY(3,K,S):: IF S<1 THEN 320
 !224
325 ! ~ START OF SCORING LOO
P FOR 3,4,5 IN A ROW !094
330 FOR TN=1 TO G :: DISPLAY
 AT(21+TN,1):N$(TN);" TEAM S
CORE IS ";TS(TN)!133
340 RESTORE 630 :: CALL TW(T
N,CN(),TS()):: FOR Z=1 TO 10
 :: CALL DW(TN,CN(),TS())::
NEXT Z !105
345 ! CHECK FOR 4 & 6 SPOT C
OMBINATIONS !004
350 RESTORE 634 :: FOR Z=1 T
O 14 :: CALL CW(TN,CN(),TS()
):: NEXT Z :: IF G>3 THEN 44
0 ! 2X2 SQUARES & 3X4 CROSSE
S (6 SPOTS) !250
360 ! CHECK FOR 8 SPOT COMBI
NATIONS !108
370 RESTORE 683 :: FOR Z=1 T
O 2 :: CALL CW(TN,CN(),TS())
 :: NEXT Z :: IF G>2 THEN 440
 ! 2X2 OCTAGON & 4X4 STAR (8
 SPOTS) !248
380 ! CHECK FOR 9-12 SPOT CO
MBINATIONS !253
390 RESTORE 690 :: FOR Z=1 T
O 20 :: CALL CW(TN,CN(),TS()
):: NEXT Z ! TRIANGLES CIRCL
ES AND K'S (9-12 SPOTS) !198
435 ! ~ END OF SCORING LOOP
!092
440 NEXT TN !056
450 DISPLAY AT(1,1):"CONTINU
E THIS GAME Y/N?" :: CALL KE
Y(3,K,S):: IF S<1 THEN 450 !
243
455 IF K=89 THEN NG=1 ELSE N
G=0 !146
460 NP=0 :: FOR S=1 TO 25 ::
 CALL MOTION(#S,RND*20-10,RN
D*25-14):: NEXT S :: GOTO 19
0 !159
470 ! ~~~ SUBROUTINE TO SET
GN(,)ARRAY FOR PLAYER !193
480 RESTORE 850 :: FOR PN=1
TO 6 :: FOR G=1 TO 3 :: READ
 GN(PN,G):: NEXT G :: NEXT P
N :: RETURN !082
490 ! ~~~~~~~~~~~~~~~~~~~~~
  DATA FOR MARKER CHARACTERS
 !186
500 DATA 49,50,51,52,53,128,
129,130,131,54,55,56,57,65,1
32,133,134,135 !140
```

(See Page 21)



TICKY-TACKY—

(Continued from Page 20)

```
510 DATA 66,67,68,69,70,136,
137,138,139,71,72,73,74,75,1
40,141,142,143 !169
520 DATA 76,77,78,79,80,136,
137,134,139 !213
530 ! ~~~~~~~~~~~~~~~~~~~~~~
 DATA FOR CHARACTERS 5AFKP
!079
540 DATA 00003F3F3C3C3F3F,00
00000030300F0F,0000F0F00000C
0C0,303030303030C0C0 !088
550 DATA 00000F0F0C0C0C0C,3F
3F0C0C0C0C0C0C,0000C0C030303
030,F0F0303030303030 !152
560 DATA 00003F3F30303030,3F
3F303030303030,0000F0F000000
000,C0C0000000000000 !224
570 DATA 0000303030303333,3C
3C33333030303030,00003030C0C00
000,00000000C0C03030 !192
575 ! ~~~~~~~~~~~~~~~~~~~~~
DATA FOR TEAM  & PLAYER PATT
ERNS !101
580 DATA 100,"FFFFFFFFF0F0F0
F0",101,"F0F0F0F0FFFFFFFF",1
02,"FFFFFFFF0F0F0F0F",103,"0
F0F0F0FFFFFFFFF" !106
590 DATA 108,"03030F0F3F3FFF
FF",109,"FFFF3F3F0F0F0303",1
10,"C0C0F0F0FCFCFFFF",111,"F
FFFFCFCF0F0C0C0" !200
595 DATA 116,"C0C0F0F03C3C0F
0F",117,"0F0F3C3CF0F0C0C0",1
18,"03030F0F3C3CF0F0",119,"F
0F03C3C0F0F0303" !206
600 DATA 58,"FFFFC3C3C3C3FFF
F",59,"183C7EFFFF7E3C18",60,
"81C3663C3C66C381" !142
610 DATA 61,"66FFFF6666FFFF6
6",62,"AAAA5555AAAA5555",63,
"FF7E3CDBDB3C7EFF" !197
620 ! ~~~~~~~~~~~~~~~~~~~~~~
  DATA FOR SCORE CHECKING
!016
625 ! FOR DIAGONAL 3,4,5 IN
A ROW !045
630 DATA 1,25,5,2,20,6,3,15,
6,6,24,6,11,23,6,5,21,4,4,16
,4,3,11,4,10,22,4,15,23,4 !0
06
633 ! FOR 2X2 SQUARES !170
634 DATA 2,4,16,0,1,5,6,1,2,
3,4,6,7,8,9,11,12,13,14,16,1
7,18,19 !126
635 ! FOR 3X3 TRIANGLES !049
640 DATA 4,6,9,0,1,2,5,6,10,
1,2,3,6,7,8,11,12,13,4,6,9,0
,1,2,6,7,12,1,2,3,6,7,8,11,1
2,13 !040
650 DATA 4,6,9,0,5,6,10,11,1
2,1,2,3,6,7,8,11,12,13,4,6,9
,0,4,5,8,9,10,3,4,5,8,9,10,1
3,14,15 !206
655 ! FOR 3X4 CROSSES !171
660 DATA 4,6,6,0,4,5,6,10,15
,2,3,4,7,8,9,4,6,6,0,3,4,5,6
,8,3,4,8,9,13,14 !094
665 DATA 4,6,6,0,5,9,10,11,1
5,2,3,4,7,8,9,4,6,6,0,4,5,6,
7,10,2,3,7,8,12,13 !184
670 DATA 4,6,4,0,2,6,10,12,1
8,1,2,6,7,4,6,4,0,2,6,10,12,
14,2,3,7,8 !228
680 DATA 4,6,4,0,2,4,8,12,14
,4,5,9,10,4,6,4,0,6,8,12,16,
18,1,2,6,7 !244
682 ! FOR 4X4 STAR & 2X2 CIR
CLE !184
683 DATA 6,8,4,0,3,6,7,11,12
,15,18,1,2,6,7 !254
684 DATA 6,8,4,0,1,4,7,9,12,
15,16,2,3,7,8 !210
685 ! FOR 4X4 BOXES !011
690 DATA 7,9,9,0,1,2,5,6,7,1
0,11,12,1,2,3,6,7,8,11,12,13
 !122
692 DATA 8,12,4,0,1,2,3,5,8,
10,13,15,16,17,18,1,2,6,7 !0
80
695 ! FOR 4X4 TRIANGLES !051
700 DATA 7,10,4,0,1,2,3,5,6,
7,10,11,15,1,2,6,7,7,10,4,0,
1,2,3,6,7,8,12,13,19,1,2,6,7
 !197
710 DATA 7,10,4,0,5,6,10,11,
12,15,16,17,18,1,2,6,7,7,10,
4,0,4,5,8,9,10,12,13,14,15,4
,5,9,10 !046
715 ! FOR 5X5 TRIANGLES !053
720 DATA 8,12,1,0,1,2,3,4,5,
8,10,12,15,16,20,1,8,12,1,0,
1,2,3,4,6,9,12,14,18,19,24,1
 !042
730 DATA 8,12,1,0,5,6,10,12,
15,18,20,21,22,23,24,1,8,12,
1,0,4,5,8,10,12,15,16,17,18,
19,20,5 !146
735 ! FOR 5X3 TRIANGLES !051
740 DATA 5,9,1,0,1,2,3,4,6,7
,8,12,1,5,9,1,0,4,5,8,9,10,1
4,15,20,5 !187
750 DATA 5,9,1,0,5,6,10,11,1
2,15,16,20,1,5,9,1,0,4,5,6,8
,9,10,11,12,3 !121
765 ! FOR 3X3 OCTAGONS !230
780 DATA 9,12,1,0,2,3,5,9,10
,14,15,19,21,22,23,2,9,12,1,
0,4,5,6,8,9,11,12,14,15,16,2
0,3 !204
785 ! FOR 5X5 CROSSES !174
790 DATA 8,9,1,0,5,8,9,10,11
,12,15,20,3,7,9,1,0,4,6,8,12
,16,18,20,24,1 !186
810 ! FOR 5X4 K AT 4 POSITIO
NS !199
820 DATA 9,10,2,0,3,5,7,10,1
1,15,17,20,23,1,2,9,10,2,0,4
,6,8,12,15,16,17,18,19,1,6 !
060
830 DATA 9,10,2,0,3,6,8,12,1
3,16,18,20,23,1,2,9,10,2,0,1
,2,3,4,7,11,13,15,19,1,6 !20
4
840 ! FOR GN(,)ARRAY !059
850 DATA 1,1,1,2,2,2,3,1,3,4
,2,1,5,1,2,6,2,3 !034
900 ! ~~ PRINT CN()ARRAY !20
5
910 OPEN #1:"PIO" :: FOR X=1
 TO 21 STEP 5 :: PRINT #1,US
ING 930:CN(X),CN(X+1),CN(X+2
),CN(X+3),CN(X+4)!237
920 PRINT #1:"" :: NEXT X ::
 CLOSE #1 :: RETURN !162
930 IMAGE ##  ##  ##  ##  ##
 !219
1000 ! #### SUBPROGRAMS ####
 !080
1010 SUB TW(T,CN(),TS())!079
1020 FOR X=0 TO 20 STEP 5 ::
 FOR Y=1 TO 5 :: GOSUB 1040
:: NEXT Y :: W=0 :: NEXT X !
077
1030 FOR X=1 TO 5 :: FOR Y=0
 TO 20 STEP 5 :: GOSUB 1040
:: NEXT Y :: W=0 :: NEXT X :
: GOTO 1060 !070
1040 IF CN(X+Y)=T+25 THEN W=
W+1 ELSE W=0 !094
1050 TS(T)=TS(T)+ABS(W=5 OR
W=4 OR W=3):: DISPLAY AT(21+
T,20)SIZE(4):TS(T):: RETURN
!079
```

(See Page 22)

TICKy-TACKY —

(Continued from Page 21)

```
1060 SUBEND !168
1070 SUB DW(T,CN(),TS())!063
1080 READ L,H,S :: FOR X=L T
O H STEP S :: IF CN(X)=T+25
THEN W=W+1 ELSE W=0 !0241090
 TS(T)=TS(T)+ABS(W=5 OR W=4
OR W=3):: NEXT X :: W=0 :: D
ISPLAY AT(21+T,20)SIZE(4):TS
(T):: SUBEND !111
1100 SUB CW(T,CN(),TS()):: D
IM A(12)!037
1110 READ V,S,C :: FOR Z=1 T
O S :: READ A(Z):: NEXT Z !1
60
1115 FOR Y=1 TO C :: READ ST
 :: FOR Z=1 TO S :: IF CN(ST
+A(Z))=T+25 THEN W=W+1 ELSE
W=0 !083
1120 NEXT Z :: IF W=S THEN T
S(T)=TS(T)+V ELSE W=0 !066
1130 NEXT Y :: W=0 :: DISPLA
Y AT(21+T,20)SIZE(4):TS(T)::
 SUBEND !189
```

Southwest 99ers take over production of Asgard Memory System

By TOM WILLS

(This article was written before the Lima Multi User Group Conference and downloaded from the Internet)

Veiled in clouds of secrecy, the Southwest 99ers met in March and April to discuss their contribution to the survival of the century. The subject was the Asgard Memory System. (Pioneers of Asgard Peripherals found themselves unable to continue the project due to the dissolving of the business; the project leader, Jim Krych, contacted us to see if we would complete the last step — actually manufacturing the product and offering it for sale.)As Asgard is no longer involved in this project, SuperAMS stands for Advanced Memory System, not Asgard Memory System.

The SuperAMS circuit card for the P-Box that replaces the 32K card and allows as little as 128K and as much as one megabyte of memory for use by the 9900 processor. This is accomplished by remapping the eight 4K segments of the 32K expansion memory space under the control of the Texas Instruments 74LS612 memory mapper device. This is the same basic technique used in the Myarc 9640 computer and the never-released 99/8 computer.

All the components are readily available on the hobby market, and there is no built-in software — no DSR is involved. Most importantly, the card is a response to a long-lived problem in the TI community —"We *need* more memory!" Because the card is fairly easy to build and answers a significant problem, Asgard Peripherals decided to release the circuit design and the software tools (the RAG assembler for AEMS, the RAG Linker for AEMS, and a collection of software libraries) to the public domain, so anyone who wanted to could build an SuperAMS. People have indeed done this, and authenticated the usefulness of the design. However, this means only those people comfortable with electronic construction would *ever* have one! To become truly useful, the card must be mass-produced and made readily available, to encourage all TI users to take the big step up in memory, and encourage programmers to provide more interesting and more capable software.

The Southwest 99ers agreed to undertake the project, with the goal of having boards ready for sale at the Lima TI fair at the end of April. With Jim Krych'a help, a professional circuit card designer has converted the circuit design to artwork for printed circuit board manufacturing. A single prototype was constructed to verify that the design was good. Unfortunately, some errors were discovered, both with the artwork and with the original design. These errors were corrected, fortunately before we paid for the entire production of faulty cards, but the time lost in recovering from this misstep means that cards will not be available at Lima. Instead, acting as our agent, Jim Krych will have the repaired prototype card at Lima, and will demonstrate its potential and take orders. He already has firm commitments from several well-known TI programmers and from members of his own user group in Ohio. Many members of the Southwest 99ers committed themselves to investing their own money in the project. Approximately $3500 is required to build the first 50 cards. At a sales price of $85 for the 128K version, $100 for the 256K version, and $25 to later upgrade the 128k version to a 256k version,, we expect the initial investors to get their money back out and for us to get a good start on the next run of cards, which will not be nearly so costly since all the engineering and setup will be complete.

Because of a good deal we got from the board manufacturer, we ordered 100 blank circuit cards; these are currently being built and we expect to receive them in early May. At the same time we are ordering parts for the cards, and when everything comes together - the boards, the parts, and the orders from Lima — we will start assembling, testing, and shipping SuperAMS cards.

As the boards are built, there will be a 24-hour burn-in test required for P-Boxes (not consoles, just the P-Boxes). The cards and parts should be received no later than mid-May and full production will begin in earnest at that time.

The SW99ers members involved in the SuperAMS project are Mike Doane, Jack Mathis, and David Ormand.

NEWSBYTES

New GROM/GRAM card released in Germany

A new GRAM card, called HSGPL, High Speed Graphics Programming Language) for the TI99/4A has been released in Germany. According to Gerd Weismann, who is promoting the card, the card allows users to remove the entire original GROM from the TI console and run the card with the faster clock timing of modules at 447 Khz. Weismann says the GPL (Graphics Programming Language) interpreter runs eight times faster on the HSGPL than without. Speed settings are software switchable.

The card has 16 GROM banks with 8x8 kilobytes (64KB) each. In addition, the card is designed with four ROM banks for each GROM bank. Other GRAM devices have only one ROM banking system. This means that only one module with one ROM can be loaded into other cards. "With the HSGPL you can load up to 16 Extended BASIC modules with their own ROM banking system," Weismann said. "Also, TI-Calc, with four ROMs and all GROM banks is loadable up to 16 times in our new card."

The card uses a one megabyte GROM and 512 KB ROM on flash EEPROMs. "These are programmable on-board and no programmer is needed." For fast access to modules running on basepage >9800, both lower GROM and ROM banks have additional memory, 128 KB of GRAM and 64 KB of RAM. The card includes an 512 KB for DSR-ROM at CRU address >1800, the location for the GPL operating system. According to Weismann, users can add commands for the GPL interpreter. Total memory on the card is 2.24 MB.

The card is offered in three configurations: Small, 1 GROM, 1 GRAM, approximately $250; Medium, 4 GROM, 4 GRAM, approximately $280; and Xpand, 16 GROM, 16 GRAM, approximately $350. Shipping is about $45 U.S. For more information, write Weismann at Koenigstr. 17-19, D-67655 Kaigerslautern, Germany.

Chicago TI fair updates plans

The Chicago TI fair, planned for October 28, promised low-cost tables for vendors and plenty of hotel rooms nearby, according to Hal Shanafield, who's coordinating the event for the Chicago TI User Group.

Hours will be from 9 a.m. to 4 p.m. in the new Evanston Public Library. Space will include a large exhibit room with a room for seminars across the hall. A banquet will follow the fair at 6:30 p.m.

There are three hotels within a two-block radius of the library, including a Holiday Inn, Omni and a local semi-residential hotel across the street from the library. There are 55 restaurants within 3 blocks. The site features plenty of free parking and is reachable via public transportation.

The price of an exhibit table is $30 to vendors. User group tables are free. The deadline for reserving tables is Sept. 1. So far, vendors RAMcharged Computers and L.L. Conner have already reserved tables for the event.

For more information, contact Shanafield at (708) 864-8644 during reasonable evening hours and on weekends.

Ink jet cartridge guide available

The Tech Press has released a reference publication focusing on the cartridges used in ink jet printers and their recycling methods. The Ink Jet Cartridge Books covers the in-depth refilling methods of 28 different ink jet cartridges. Also included in the book is a cross reference and resource guide.

The book sells for $39.50. For further information, contact The Tech Press, 3107 W. Colorado Ave., Colorado Springs, CO 80904 or 1-800-670-2224.

Send your product and service announcements to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680.

1995 TI FAIRS

APRIL

Lima Multi Users Group Conference, April 29, Reed Hall, Ohio State University at Lima. Contact Lima Users Group, P.O. Box 647, Venedocia OH 45894, or call Charles Good (evenings) at (419) 667-3131 or Internet cgood@osulima1.lima.ohio-state.edu.

SEPTEMBER

10th International TI-Meeting, Sept. 22-24, Wohlfahrtsgebäude der Wiener E-Werke (Welfare Building of the Vienna Electricity Board), Wachaustr. 28, A-1020 Vienna, Austria. For information write Kurt Radowisch, TI- and Geneve User Group Vienna, Fugbachgasse 18/17, A-1020 Vienna, Austria.

OCTOBER

Chicago International TI Faire, October 28, Evanston Public Library. Contact Hal Shanafield, (708) 864-8644.

1996 TI FAIRS

FEBRUARY

Fest West '96, Feb.17, Quality Inn, 1601 Oracle Dr., Tucson, Arizona. Contact SouthWest Ninety Niners User Group-by sending e-mail to twills@primenet.com. Or call the Cactus Patch BBS at (520) 290-6277.

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.

MNP tips

Modem users may find an easy performance boost in this protocol

The following was written by Jerry Coffey. We found it on a BBS. While some TI users may be using 9600 baud or faster modems, many others are using 2400 and 1200 baud modems. This may help them reduce the cost of telecommunicating.—Ed.

MNP stands for Microcom Networking Protocol and is actually a series of nine separate protocols (called levels) developed by Microcom to improve the performance of their modems. In a marketing masterstroke, Microcom licensed Levels 1 through 5 to other modem manufacturers for a reasonable fee. This open policy has made MNP a de facto industry standard. Even such heavyweights as U.S. Robotics support it in their high speed line along with the international CCIT standards (e.g., V.32). The most telling indication of the Microcom influence in the marketplace is the support of the common MNP protocols by Compuserve, GEnie, and TELENET.

HOW IT WORKS

The three most popular MNP Levels (numbers 3, 4 and 5) all use a type of quasi-synchronous transmission. Normal asynchronous communication paces its transmission at the byte level by "framing" each byte with a start bit and a stop bit. Synchronous transmission, as its name implies, uses a synchronized clock rate at both ends of the line which allows individual bits to be sent at a constant rate, avoiding the need for the extra two bits per byte. This allows synchronous transmission a 25 percent greater throughput on any given channel. MNP uses the same idea by sending relatively long bursts of data bits between the bits that control timing, thus reducing the overhead of normal "framing" bits (at least in theory). MNP level 5 includes a data compression algorithm which effectively doubles the throughput of raw data. This is only useful for uncompressed data, mainly long ASCII transfers such as nonstop message dumps since most long binary files are already compressed when they are transmitted.

MNP modems usually have on-board RAM to serve as a buffer between MNP mode transmissions and the asynchronous link between the modem and your computer. My MNP level 4 modem had 2K of static RAM while the level 5 version has 8K to buffer the decompression process. This means that the modem continues to receive while your computer is writing to disk. To take full advantage of the modem, you must set the computer-to- modem link at a faster bps rate so it can empty the buffer faster than it is filled. Otherwise the transfer will slow down as the buffer fills.

Actually the MNP protocols have substantial overhead of their own that offsets some of their gains. One source of overhead is the extra bytes dedicated to error checking (not unlike the CRC or checksum attached to an Xmodem packet of 128 data bytes). The other source of overhead is changes in the direction of transmission — the protocols work most efficiently for long streams of transmission in one direction. It appears that Compuserve is set up to minimize overhead when the host is sending — I got lots of timeouts between blocks trying to send TO CIS, but haven't checked this yet on other networks.

FAST ASCII WITH MNP

In a previous article I noted that MNP could increase transmission of ASCII material above the theoretical maximum rate for asynchronous transmission (this maximum is about 240 bytes per second for a 2400 bps channel since each byte takes 10 bits). The reason this asynchronous maximum can be exceeded is that MNP saves the time consumed by the two extra "framing" bits for each byte. The 2400 MNP

(See Page 25)

Comparison of transfer rates

	Compuserve	Delphi	GEnie
1200 A	115	115	115
2400 A	220	220	—<— not reliable - data loss
2400 AM	227	nt	242
1200 X	90	75	65
2400 X	145	125	120
2400 K	205	140	145**
2400 Y	210	140*	—<— locks up
2400 C	205	NA	NA

A=ascii
AM=ascii+MNP
X=xmodem
K=1kxmodem
Y=ymodem batch
C=CISB+
K,Y, and C are all under MNP
NA means Not Available
nt means not tested
— indicates a setup that did not function properly

* Delphi uses default filenames which speeds up Ymodem batch beyond the effect of the indicated transfer rate

** GEnie's instantaneous rate slightly exaggerates the actual throughput, since GEnie's 1Kxmodem pads the last block out to 1024 bytes which adds time to the download for the filler bytes.

MODEMS—

(Continued from Page 24)

maximum rate is 300 bytes per second less the overhead due to burst synchronization, error-checking and transmission reversal. At level 4, this can give actual ASCII throughputs greater than 240 bytes per second — level 3 is slightly slower but near the 240 bytes per second rate.

With Xmodem transfers, this advantage is lost. The need to turn around ("ACK") every 128 bytes slows the transfer below the rate of a straight 2400 bps download. Given the poor Xmodem performance, I did not expect much of any binary transfer.

AND NOW FAST BINARY

I used an alpha version of a terminal emulator, which includes some large-block transfer protocols (initially 1K xmodem). As the program has developed, Ymodem (batch) and CISB+ have been added. While logged onto Compuserve using MNP to test ASCII dumps, I decided to try a 1Kxmodem download.

I was amazed at the 200+ bytes-per-second transfer rates I saw!

Since that experience I have tested one or more of the large-block protocols under MNP on Compuserve, Delphi (via TELENET), and GEnie. The results appear below along with my normal experience (for reference) on each of the local Washington, DC nodes that I use regularly. *(The author used Telco in some of his tests.—Ed.)*

What I found was that) 2400 bps with MNP and large-block protocols provides a distinct speed and cost advantage on all three networks.

As noted in previous articles, the difference in price must be considered to determine the amount of savings. What I found was that) 2400 bps with MNP and large-block protocols provides a distinct speed and cost advantage on all three networks. *(The author includes a chart comparing the various services, but prices have changed and including the chart here would be meaningless.—Ed.)*

On all the networks there was one less than obvious advantage — the built-in MNP error checking which operates within the large blocks and virtually eliminates retries. This is important since retransmitting an entire block of 1024 bytes can add considerable time to a download.

HOW TO USE IT

To make all this work, you need to set your terminal program to 4800 bps (remember that the link between your computer and your MNP modem is buffered and is still operating in asynchronous mode). Send the setup string for MNP mode to your modem. Then log on to the network through a 2400 bps MNP node. The modem will automatically set up the MNP phone link, indicate "CONNECT 2400/REL", and you are on your way.

BOTTOM LINE

The message should be clear by now: use a 2400 bps modem with MNP. You don't need the more expensive Level 5 unless a local BBS supports it (many do) — the cheaper modems with MNP Level 4 will allow you to connect through existing nodes on all three of the big networks.

BUGS AND BYTES

SuperAMS schematic error

Jim Krych posted this item on the Internet:

After some work with the SUPERams schematics. An error was discovered. This error is only of concern to those NOT using PEB Protoboards-but making their own board from scratch.

The Data transceiver pins on the schematic are switched. "B" pins should be the input from the edgecard bus, "A" pins should be the output. If you have made the design like this, a simple fix is to take the -DBIN from pin 8 of 74ls04 and run it to pin 1 of the '245. This will take care of the problem.

This was not for PEB Protocard users. However, the schematics are being redrawn to clear this up.

I am sorry and publically apologize for any inconvenience.

San Diego users do good deeds with TI

The Southern California Computer Group, located in San Diego, has a couple of community educational programs going on, both coordinated by member Marti Alvarez, who is a newly certified teacher.

The group is collecting TI computers and modules for teaching the beginning levels of reading, writing, beginning English vocabulary and the sounds of the English language. These are to be used with Kurdish refugees from Iraq in the San Diego area. The SCCG is looking for a learning center to set up the computers. Meanwhile, the group is collecting TI preschool and kindergarten programs on videotape. (Educational programs may be donated to the SCCG, P.O. Box 152535, San Diego, CA 92195, or contact Alvarez at (619) 440-1209.

TI computers are also in use at a program in El Cajon, California, sponsored by the San Diego County 4-H. This is an extended before and after school program at school sites. TI computers are in place at four of the seven sites for the program, with monitors pending for the other three.

MICRO-REVIEWS

PFM+ and Board Games

By CHARLES GOOD

I now have a Geneve and am able to review Geneve specific software and perhaps hardware that anyone wants to send my way. I will of course continue to review 99/4A products.

PFM+ by Cecure Electronics

A MICROpendium article from November 1989 about the Myarc Geneve says, "Concerning an EPROM based version of DOS, (Myarc's owner Lou) Phillips said that few users are interested in it. Such a system wouldn't have the memory to support hard disk drives nor could it be updated easily." My, how times have changed. Everything about Lou's statement is now incorrect.

One of the new computer technologies, programmable flash memory, is described by Tony Lewis in the May 1990 issue of MICROpendium. To reprogram a flash EPROM you don't need any special hardware and you don't have to remove the flash EPROM from its circuit board. Recently Cecure Electronics has given Geneve owners access to this technology in the form of a reprogrammable on board flash EPROM containing the Geneve's operating system. Called PFM, this product is described by Cal Zanella in the June 1994 MICROpendium. PFM consists of a 128K flash EPROM chip that replaces the boot EPROM that came with the Geneve. My PFM flash EPROM came preloaded with the latest non SCSI version of MDOS SYSTEM/SYS and this will load automatically after about 15 seconds from a cold boot and almost immediately from a warm boot. If you temporarily don't want to use the operating system in the PFM chip you can, with a keypress, load an alternate operating system from any device. If you want to put an upgraded operating system in the PFM chip that is also very easy to do.

PFM has one significant limitation. It is only 128K. This is too small to hold the SCSI version of MDOS, which takes up 136K. Cecure is hoping to have a data compression system that will allow such large SYSTEM/SYS files to be loaded into PFM, but as of this writing you can't load SCSI-compatible MDOS into PFM. If you already have PFM you have to hit the space bar as the Geneve powers up, and then select the menu Floppy or Horizon RAMdisk as the source of the SCSI version of MDOS. If you plan to use the SCSI card you may not want to purchase PFM. However, you may want PFM anyway so that you can have the benefits of PFM+. My, how things have changed. The last versions of MDOS officially mailed out by Myarc took only 90K and would fit on a SSSD disk.

Now Cecure has taken flash memory technology one step further. Cecure now offers to install a reprogrammable flash simulated disk drive right on the Geneve board where you can keep your most important software. For those early Geneve owners who had to get used to booting from floppy or figuring out how to configure a Horizon RAMdisk or hard drive to boot the operating system, the new Cecure Flashdisk system is almost like magic. "Look carefully. There's nothing up my sleeve, no RAMdisk or hard drive controller or memory expansion card in the P-box and no floppy in the drives. Now watch the magic as I turn on the P-box and the Geneve loads MDOS. I now type'DIR DSK8.' and you can see that I have immediate access to all these neat files on drive DSK8, 509 sectors of files."

This PFM+ Flashdisk is a second 128K flash memory chip Cecure installs on the Geneve board. It can be reprogrammed thousands of times to suit your needs. It requires no battery, no P-box card slot, no CRU address, and has no DSR. It does however require that you already have PFM, the flash chip containing MDOS. The Flashdisk is piggyback soldered on top of the PFM chip so you now have a layer of two chips in the socket once occupied by the Geneve's original EPROM. Dealer installation of PFM an PFM+ is required. There is as yet no do-it-yourself kit.

The Flashdisk is normally used as a ROM disk. Right now I have the latest 80-column Funnelweb disk review and editor with its 128K text buffer on my Flashdisk. I am using the Funnelweb word processor to write this article. Any time I want I can exit the word processor and go to Funnelweb's disk review to examine the contents of my Flashdisk. Using disk review I can examine sectors, get a Flashdisk directory (drive is DSK8, volume name is FLASHDISK) and view or run files directly from this directory in exactly the same manner I would deal with files on a floppy or hard drive. The Flashdisk responds in the expected way to any disk management software. But don't throw away your Horizon RAMdisk yet, because the Flashdisk has some limitations. What I cannot do in the normal way is write to the Flashdisk. When I want to save this article, I will have to save it to a floppy (or a Horizon or a hard drive). I can't go to the word processor's command line, press SF (save file), specify DSK8, and save this text to my Flashdisk. Using normal disk management software I also can't delete or upload files to the Flashdisk, sweep or reformat Flashdisk, or change its sectors with a sector editor.

When you reprogram a flash memory chip you have to reprogram the whole thing all at once. You can't write to just part of the memory on a flash memory chip. To program the PFM+ Flashdisk you create a 128K ramdisk in the Geneve's memory and temporarily place on this RAMdisk all the files you want to upload to the Flashdisk. You then use special software provided by Cecure to convert all the files you put on the RAMdisk into one great big file which you save to any mass storage device. You then reboot the Geneve and quickly hit the space bar. At this point you are presented with a menu that is

(See Page 27)

MICRO-REVIEWS —

(Continued from Page 26)

part of Cecure's Geneve boot BIOS v3. This menu gives you the opportunity to load a new operating system (MDOS) into PFM, temporarily boot an alternate MDOS without changing that resident in PFM, or reprogram the Flashdisk. Press L to (L)oad the Flashdisk and select from the next menu the kind of mass storage where you are keeping your great big file. If you have several of these great big files on line you can select, with a single keypress, any one of 11 such big files to load the Flashdisk. Finally, press Y to confirm that you really do want to reprogram the Flashdisk. That's all there is to it.

Using only on-board flashdisk files you can load and run from disk any 99/4A or Geneve specific software....

If you really wanted to do it, you could run games on the Geneve with only the software on the PFM+ and have the Geneve be the one and only card in the P-box. No disk controller, no hard drive controller, nothing. For serious computing there are enough sectors on the PMF+ Flashdisk to contain GPL, the Extended BASIC module files, the Editor/Assembler module file, EXEC and HyperCopy, with a little room left over for some Autoexec files. Thus, using only on-board Flashdisk files you can load and run from disk any 99/4A or Geneve specific software and you can do some really fast disk copying.

There is one other little thing. Everytime MDOS starts device T has to be remapped as a drive number in order to access the Flashdisk files. I like DSK8. You can do this in several ways. The most sophisticated way is to use Cecure's CYA software to permanently alter MDOS so that device T always automatically maps as your choice of drive numbers. You can also type REMAP 8T from the A> prompt, or you can put this statement in an autoexec file. Following this with ASSIGN F=DSK8: makes it even easier to access files on the Flashdisk.

PFM costs $75 installed. If you already have PFM then PFM+ costs $60. The cost for PFM and PFM+ installed at the same time is $125. These prices include the latest MDOS and software needed to reprogram the Flashdisk.

BOARD GAMES By Jimmy Dowell

This disk contains five games and is designed to be run from DSK1. You get an introductory menu from which you can load any of the games, and you can also from this menu print the instructions to any game to the screen or to a printer. When a game ends, or when you press the designated key to prematurely abort a game, you are returned to the introductory menu. These are not arcade fast action games. In fact, none of them use a joystick. They are instead strategy games resembling Barry Traver's "Coney Games" disk released a few years ago. Some of the games use computer-generated dice rolling to produce random numbers used in game play.

Technically the games are interesting. Screen displays are excellent and colorful. Most games are accompanied by assembly language music which continues to play even if you go into command mode to examine the games' coding. The only way to stop the music is to OLD another program. Speaking of command mode, there are no ON BREAK NEXTs in these games. This means that you can at any time back out of the game with FCTN/3 and enter command mode. Some programmers consider XB software made unbreakable with ON BREAK NEXT to be an example of a professional sophisticated programming technique. I think ON BREAK NEXT is a big nuisance. I like to go into existing programs and alter them a bit, adding my printer's NLQ font coding for example. This hard to do with an ON BREAK NEXT in the program.

Game titles include Can't Quit, Nothing But Trouble, Peg Jump, Fox Hunt, Peg Jump, and Backgammon. My favorite is the one-player game Peg Jump which I actually played on a plastic game board with holes in it in my youth. The object is to jump pegs, removing the jumped peg, in such a way as to leave only one peg on the board in the center hole. Years ago, all by myself, I figured out one of the several possible perfect solutions.

Fox Hunt is somewhat like Peg Jump. In this one or two player game the single fox moves by jumping over the hounds, removing the jumped over hound. Then the hound player can move one of his pieces. The object is to get the fox cornered so it can't make a legal move before too many of the hounds are eliminated.

Can't Quit is hard to describe. Each turn starts with a roll of four dice, from which the player picks one of several possible pairs of dice numbers. You place markers on the board, based on these numbers and then roll again. When you can't make a legal move your turn ends.

Backgammon is, well, Backgammon. In Nothing But Trouble from one to four players move all their several men around the board from start to home based on the roll of dice. Sometimes men get sent back to start and have to begin again.

Board Games comes on a SSSD disk and is fairware. The author doesn't specify a donation. I think maybe $5 would be appropriate. You can obtain Board Games by sending a fairware donation a disk and a paid return mailer directly to the author, or you can send me $1 to pay for the disk and postage and my vacation fund and I will send it to you.

ACCESS

Cecure Electronics (send them your Geneve for installation of PFM+), P.O. Box 132, Muskego WI 53150. Credit card orders phone 800-959-9640. Voice information 414-679-4343. Fax 414-679-3736.

Jimmy Dowell (Board Games author), 2310 Adams Ln. #68, Henderson KY 42420, phone 502-827-5812

Charles Good, P.O. Box 647, Venedocia OH 45894, phone 419-667-3131, internet email cgood@osulima1.lima.ohio-state.edu or good.6@osu.edu

USER NOTES

Corrections to Forget-Me-Knots and Reminders

This comes from Ron Warfield, of the British Columbia 99er User Group. He writes:

If you are using Alfred Malcom's Forget-Me-Knots and get a syntax error when trying to quit from the menu, here is the fix:

At the beginning of the LOAD program on line 2, add CALL INIT to the list of CALLs. Somehow this CALL got erased and was not noticed because the computers we tested the program on had GRAM Krackers installed. You do not need a CALL INIT with a GRAM Kracker. Here is the fix:

```
2 CALL INIT :: CALL CHAR ::
 CALL CLEAR :: CALL COLOR ::
 CALL HCHAR :: CALL KEY :: C
ALL LOAD :: CALL SCREEN :: C
ALL VCHAR :: DIM A$(12) :: !
@P-
1000 CALL INIT :: CALL LOAD(
-31962,8)
```

Also, in the program Reminders, if you have an MBP clock you will have to change line 180 to read J$="95" so date is correct. Also change line 400:

```
400 CALL INIT :: CALL LOAD(-
31961,51) :: END
```

We also would like to correct the KEY statement in line 950 to read:

```
950 CALL KEY(@,D,E)
```

The line originally was CALL KEY(_,D,E).

This was reported in the February MICROpendium but was taken from our formatted newsletter on disk, so an extra @ was added.

Correction to INTLEARN, INTINDEX

This comes from Phil Van Nordstrand of the Houston Texas Users Group. He writes:

This is just a short note to point out the problems with the INTernational programs on the March 1995 disk. *(The same errors appeared in the listings in MICROpendium.—Ed.)*

INTLEARN would not display the English words on the left side of the screen. Looking at the file INTNDX showed that only space characters showed up, accounting for the problem.

Looking at INTINDEX, an error showed up in line 150. It should start out with IN$=IN$&R$ :: , etc. Running this fills in the English words on the screen and in the file INTNDX. Then INTLEARN will work as advertised.

WORD* is obviously incomplete, as only 5 lines out of 100 data lines were typed in, as mentioned in the REM statement in line 1006. This file should be saved in MERGE format and then merged with INTDATA. This program won't run and it seems that lines 160, 170 and 180 are all suffering from a lack of ampersands.

Line 160 should end with .... THEN W$=" "&W$

Line 170 should start with W$=W$&RPT$(etc.

Line 180 should start with R$=R$&W$ :: etc.

These omissions are clear from the arti

(See Page 29)

Lima conference videotapes

Below are the directories of the three videotapes of the recent Lima Multi User Group Conference. There are approximately 16 hours of viewing, with some title screens created by Bruce Harrison and displayed using his Video Titler software. Charles Good already has orders for about 25 sets of tapes and will take orders from anyone who wants a set.

To order, send a check for $15 to Good, P.O. Box 647, Venedocia, OH 45894. (Good can also be reached at cgood@osulima1.lima.ohio-state.edu. The price includes the tapes and book rate postage within the U.S. "Please send a little more money for airmail delivery to foreign locations," Good said.

CONTENTS OF THE LIMA CONFERENCE VIDEOTAPES

1995 LIMA TI CONFERENCE—Tape 1 Table of Contents

SPEAKER	TOPIC)	(Tape counter)
Barry Traver	The TI in a PC world	125
Bob Carmany	Making & dumping eproms	1350
Charles Good	Funnelweb v5.2 80 col editor	2675
Jim Krych	Super AMS mem expansion for /4A	3500
Bud Mills	Demo of SCSI hard drive Geneve	4425
Ron Markus	Asgard software, Prostick, etc.	5140

1995 LIMA TI CONFERENCE—Tape 2 Table of Contents

SPEAKER	TOPIC)	(Tape counter)
Mike Maksimik	Geneve, SCSI, tape backups	100
Tim Tesch & Maksimik	PORT demo on Geneve	1500
Don Walden & Tesch	CYA,PFM+,Geneve upgrades	2850
David Nieters	Demo SCSI &AT keyboard on 4A	3900
Bruce Harrison	Titler & other PD software	4525

1995 LIMA TI CONFERENCE—Tape 3 Table of Contents

SPEAKER	TOPIC)	(Tape counter)
Mike Wright	PC99 stage 3	100
User Group Officers' Conference		1650
Mary Phillips	Artist Card Shop	2800
Charles Good	RXB v1001, and Term 80	3850
John Bull	Windows for XB	4350

USER NOTES

(Continued from Page 28)

The listings where the ampersand has been replaced with a tilde or some such character. Probably from use of the TI-Writer formatter or use of transliteration. I ran the programs after making the corrections and merging the data. It bombed out due to lack of data, but I "CLOSed #2" file and looked at it with a sector editor. The file looks okay.

Back trouble? What's afoot?

Many back pains and posture problems experienced by persons who sit at computers can be traced to their feet, according to Nancy Osterman, director of the Workplace Ergonomics Institute at Quill Corporation.

Persons whose feet are off the floor while typing are candidates for muscle aches and pains, she says. But the solution isn't pills, it's a footrest.

"Dangling your legs off the front of a chair increases pressure on the backs of thighs and reduces circulation of blood," Osterman notes. "In addition, your hips and spine have more pressure placed on them because the body's weight isn't even partially supported by your feet."

She adds, "When feet are supported on a footrest, a shorter employee can sit back and fully utilize the backrest for support and comfort. Even taller people, who can firmly plant their feet on the floor, will find that a footrest adds another position for their feet and helps eliminate fatigue in the back and legs."

Shake hands with a null modem

This item appeared in the newsletter of the Dallas TI User Group (Dallas 99 Interface). It was written by Jim Stewart.

I never met a null modem I didn't like, or understood, for that matter. I have used them, bought them, read about them, tried to understand them. But I've never had a clue what a null modem was.

Recently I found in *The Brady Guide to Microcomputer Troubleshooting and Maintenance* a paragraph that puts some light on the subject. Before sharing these illuminating words, I should try to explain how I relate to this confusing subject.

My hearing is so bad that 50 percent of my conversations consist of "What's that?" Not that I can blame her, but my wife, Lynne, rarely speaks to me head on. Instead she will start talking into the cupboard while cooking dinner and I'm reading a book 40 feet behind her. If I can see that she is talking to me, my comprehension goes way up. Or, to say it another way, we talk towards each other but we don't communicate because our messages don't "shake hands" or determine if the

(See Page 30)

USER NOTES

(Continued from Page 29)

line is busy. Now I have figured out that Lynne is the DTE (Data Terminal Equipment) and I'm the DCE (Data Communication Equipment) in our family, I come a lot closer to understanding how Mr. Henry Beechhold describes RS-232 connections.

"The confusing part of (RS-232) is the terminology, TRANSMIT and RECEIVE, you have to think of the terms this way. In DTE, TxD is outgoing and RxD is incoming, but in DCE, on the other hand, TxD is incoming and RxD is outgoing. A cross-wired arrangement to achieve the desired mating is called a *null modem*. If you wanted to transfer files between two computers in physical proximity, you could do so by connecting the two serial ports with a null modem cable, forcing one of the computers to act as DCE to the other's DTE."

There you have it. It's really simple when it's explained that way. Right?

Oh oh! Here comes the DTE looking for the DCE to do some WORk. I'm out of here.

LOWERCASE corrected, LABELER3 clarified

The following comes from Dave Swartz of Aurora, Colorado. He writes:

In running the programs on the March disk, I found a problem with the LOWERCASE program. I continually got a "BAD VALUE in 1" when trying to run the program. After much examination and testing

(See Page 31)

READER TO READER

Bruce Harrison, 5705 40th Place, Hyattsville, MD 20781, (301) 277-3467, writes:

The reader telephone request MICROpendium relayed to me isn't very detailed, so it's hard to figure out exactly what's desired. If he or she wants a way to fill in a rectangular area on the graphics screen with some color, the source file FILCOL/S should be a big help. This shows a way (one of perhaps a thousand ways) to fill in an area with any chosen color. In this case, we fill an area starting at row 11, column 6 and extending 10 rows and 17 columns from that point with solid magenta.

We start by selecting a character that's not being used otherwise. In this case, we've used character 128. The color byte for that character set is at >390 (>380+16=>390) in VDP RAM. We set the left byte of R1 with>DD00 so that the foreground and background are both magenta, then BLWP @VSBW to put that color in place. (All the characters from 128 through 135 will be solid magenta.) This way, it doesn't matter whether char 128 has a defined pattern.

Now, we do a nested loop that writes character 128 repeating 17 across by 10 down, thus filling an area with magenta. When that's done, we just enter a key loop so that the user can see the color block on the screen. Pressing any key will exit the program.

```
* FILL COLOR
* FILCOL/S
*
* FILLS A SCREEN AREA 10 X 17 SPACES
* WITH MAGENTA COLOR
*
    REF VSBW,KSCAN REF UTILITIES
    DEF START        DEFINE ENTRY
WS EQU >20BA         USER WORKSPACE
START LWPI WS        LOAD USER WS
    LI R1,>DD00      LEFT BYTE OF R1 HAS DESIRED COLOR TWICE
    LI R0,>380+16 COLOR FOR CHARS 128-135
    BLWP @VSBW       PUT COLOR IN PLACE
    LI R9,10*32+5 STARTING POINT ROW 11, COL 6
    LI R4,10         TEN ROWS
    LI R5,17         SEVENTEEN COLUMNS
    LI R1,128*256 CHARACTER 128 IN LEFT BYTE R1
OUTLP MOV R5,R2      COPY COLUMN COUNT IN R5 INTO R2
    MOV R9,R0        PUT START POSITION IN R0
INLP BLWP @VSBW      WRITE ONE CHAR
    INC R0  NEXT COLUMN
    DEC R2  DEC COLUMN COUNT
    JNE INLP         IF NOT ZERO, DO ANOTHER
    AI R9,32         DOWN ONE ROW ON SCREEN
    DEC R4  DEC ROW COUNT
    JNE OUTLP        IF NOT ZERO, DO ANOTHER ROW
    CLR R1  ZERO IN REG. 1
KEY BLWP @KSCAN      SCAN KEYBOARD
    LIMI 2  ALLOW INTERRUPTS
    LIMI 0  STOP THEM
    MOVB @>837C,R1 COPY GPL STATUS BYTE INTO R1
    CI R1,>2000      COMPARE TO >2000
    JNE KEY IF NOT EQUAL, SCAN AGAIN
    LWPI >83E0       LOAD GPL WORKSPACE
    B @>6A  GO TO GPL INTERPRETER
    END
```

USER NOTES

(Continued from Page 30)

I finally compared a printout of the disk program with the program printed in your magazine. There it was — the disk copy had three spaces inserted into the DATA statements in lines 2, 3 and 4, and one space in line 5. See attached printout. Deleting the blank spaces resulted in the program running as designed.

The January disk also contains an error, at least on my copy. Line 420 reads:

```
420 CALL RESETPLT :: P<Q=f4V3sHKS etc.
```

This program gives me a "CAN'T FIND SUBPROGRAM IN 100" error when trying to run it. As near as I can tell, there is no subprogram named GRAPHICS in the program. Is a line missing?

There is no missing line. The line reads:

```
420 CALL RESETPLT :: CLS :: END
```

The listing on page 17 of the January MICROpendium is correct. This is Jim Uzzell's LABELER 3 program, which requires Myarc BASIC. The line is correct in the original program when listed using Myarc BASIC. When listed using TI Extended BASIC it contains apparent errors. That's because XBASIC doesn't know how to interpret CLS.—Ed.

MICROpendium pays $10 for items sent in by readers and used in this column. Send them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

CLASSIFIEDS

FOR SALE

BUY-SELL-TRADE

Hardware — Software. TI Buyers Guide $2. Dual 1/2 HT Kit complete with cab les, screws, drill template & instructions $59. PE Box complete $149. Corcomp DD Cont $180. Star NX-1001 Printer $175. 2400 Baud Modem $79. TI99/4A Console $45. Rare Myarc Hard & Floppy Controller with 10 Meg Hard Disk & Case $350. 800-471-1600 (Nationwide & Canada ORDERS ONLY). 414-672-1600 Local/tech Support. Huge Genuine TI Inventory Since 1982. Competition Computer, 2219 S. Muskego, Milwaukee, WI 53215. Fax 414-672-8977. Open Daily 9-5, Sat 10-3. Bankcards, Discover, Checks & UPS/COD.

T.O.D. FANS!

All new Tunnels of Doom Adventure for your TI! A major revision with new graphics, spells, ploys, and surprises! It's called **Halls of Lost...MORIA!** and takes place in the dawn of the Fourth Age. Listen to these rave reviews: Justin V (age 10) Neat! It looks more like a dungeon, the fireballs are awsome! Amy V. (age 6) I think it's pretty! Diane V. (age ??) Hmm... if you cut wood, like that Dwarf chops Orcs, we wouldn't be worried about running short this winter.... Send $4.95 ($5.95 outside USA) for disk and booklet to: Michael Veprauskas, 32 Vaillancourt Dr., New Ipswich, NH 03071. v12n5

TI99/4A

TI software & hardware. Joy Electronics Inc., P.O. Box 542526, Dallas, TX 75354-2526. 214-243-5371. Call or write for free catalogue. v12n5

PROGRAM INNOVATORS SOFTWARE SALE

TOUCHDOWN: NFL Football Predictor $10.00 WALLSTREET: Investment Package $30.00 USVBA Power Volleyball (ML-Game) $10.00 Klingon Invaders, Destroy Klaatu, Desert Rat (Machine Language Games) $10.00 Cutthroat Cribbage, Titris, Cockroaches, Snomobile, Martian Missiles, $10.00 Multiple Orders 10% discount 4122 Glenway, Wauwatosa WI 53222-1116.

TIPS

It took over 340 hours but now all TIPS files have been categorized by subject. You can now locate any picture in seconds. Duplications have been eliminated and those crazy file names are history. Printable Index is included. Available on S/S OR D/S (specify) brand new disks, with sleeve and disk label already printed. Disk copying time will vary depending on responses. Each club should have this treasure so members may copy and reimburse your cost many times over. Send Check or MO for $50.00 to Raymond Frantz, 502 N. 51st Street, Apt. #2, Phoenix, AZ 85008-6649 v12n5

LOOK

Classified ads are 10 cents per word

If you want to sell or buy something,
advertise it in MICROpendium Classifieds.
Simply write your ad on a separate sheet of paper,
count the word (a phone number counts as one word)
and send it, along with payment, to:

MICROpendium Classifieds,
P.O. Box 1343
Round Rock, TX 78680.

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, $35 ❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50** ❑ 12 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. List issues:____________________

____

No price breaks on sets of back issues. Free shipping USA. Add 30 cents, single issues to Canada/Mexico. Other foreign shipping 50 cents single issue surface, $1.50 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