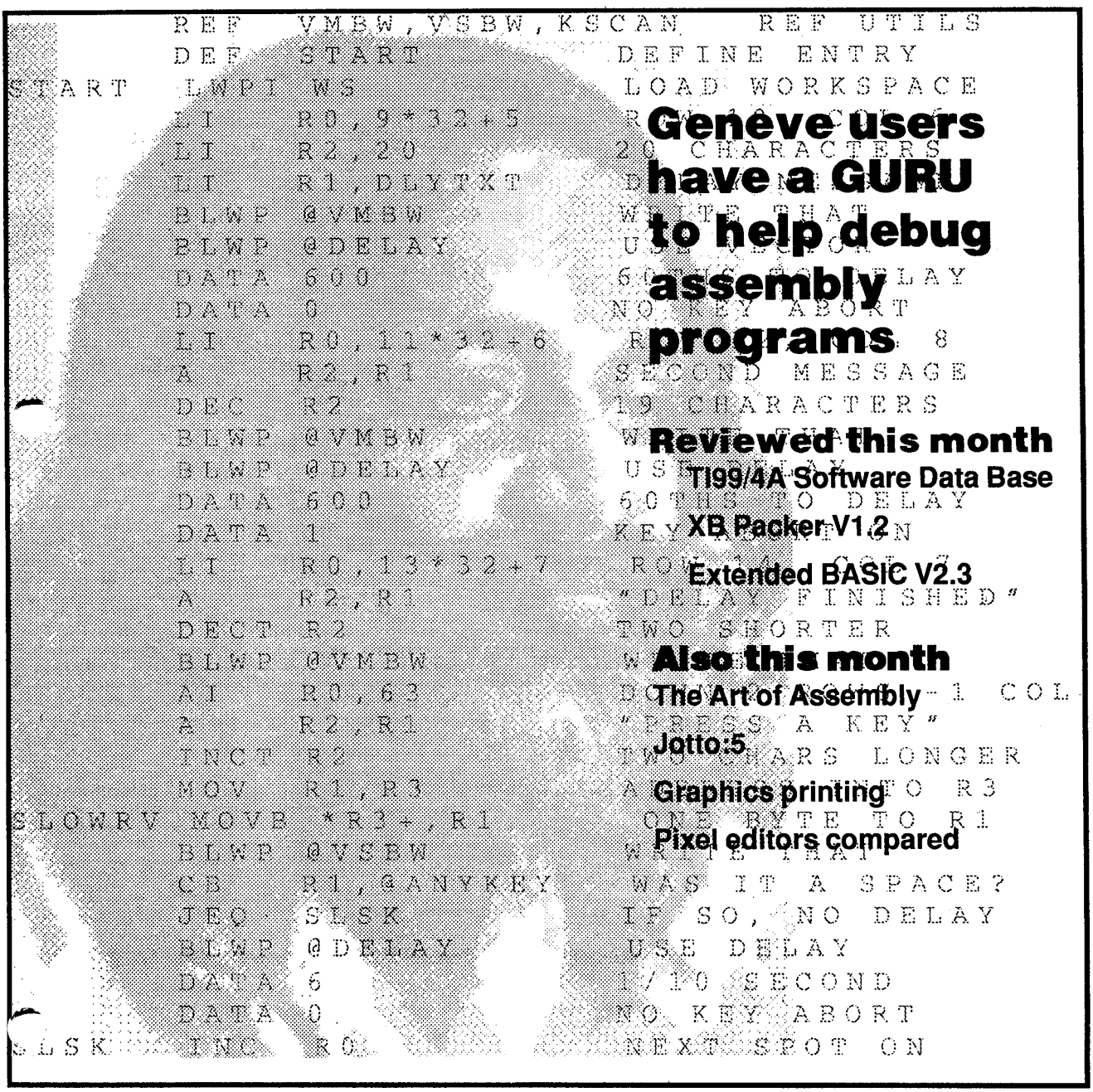
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 7 August 1995 $3.50



Geneve users have a GURU to help debug assembly programs

Reviewed this month

TI99/4A Software Data Base

XB Packer V1.2

Extended BASIC V2.3

Also this month

The Art of Assembly

Jotto:5

Graphics printing

Pixel editors compared

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.

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.

TEX COMP LTD.

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

CAN'T FIND DISKETTES FOR YOUR TI ?

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

**BULK DISKETTES*

SUPER BUY!

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

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

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

CALL FOR QUANTITY DISCOUNTS

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

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

DISKS BY THE BOX

ADD .01 ADDITIONAL SHIPPING PER DISK

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

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

DBI Data Bio Tics

SOLID STATE CARTRIDGES

SUPER AMS CARD

TAKE YOUR TI TO THE NEXT LEVEL!

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

There are two versions to chose from

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

CALL FOR DETAILS

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

TI WORKSHOP $49.95

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

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

ORDER BY PHONE

VISA & MASTER CARD EXCEPTED

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

ADD 5% SHIPPING {$5.00 MINIMUM}

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

COMMENTS

Coming around the circle

Does the deterioration of age strike magazines, as well as people? When you're young, you celebrate every birthday. When you're older, you don't. It's not that you're ashamed, it's just that you don't see it as something to celebrate. Instead, each passing year is a mile-marker on the road toward mortality.

Try as you might, you can't keep the years from catching up with you. Deterioration occurs. You first see it in an increase in aches and pains, then your joints start to bother you, and then, well, it's different for everyone.

In some ways, the same process holds true for MICROpendium. When we started this magazine, we did it after work, on nights and on weekends. Then it became a fulltime job for both of us, with other publishing endeavors on the side.

Now we have come full circle. We have gone back to work, and we are producing MICROpendium after hours.The difference is that now we don't envision, as we once did, an expansion to something bigger. But we do hope to continue it as long as a core group maintains its interest in the TI99/4A.

A POSITIVE NOTE

We're back to 32 pages this month and should remain that way for the foreseeable future. **—JK**

FEEDBACK

Program available in new version

I recently received a request for help with one of my TI programs for which source code was listed in MICROpendium (FREELOADER III, Not your average load program, February 1995).

This was quite a surprise for me, since I had no knowledge of the article and have yet to see it! Although I am delighted to contribute to the TI community, I feel that I should have been notified prior to publication. If I had been, I would have provided a correct address and the source code for FREELOADER VI, which is much faster than version III. I might also have added explanations concerning the use and understanding of the program.

A few of my programs have shareware notices with my old address. If you encounter one of these, please disregard the shareware plea, as I now consider all of my TI programs to be in the public domain.

Karl Romstedt
2330 Edgevale Rd.
Columbus, OH 43221-1218

We downloaded the program from a BBS. Time constraints made it impractical to contact the author, though a letter should have been sent even though it would probably not have been delivered as the only address we had was old. We have since mailed the author a copy of the February edition.

Issue disappoints

I just received the July 1995 issue of MICROpendium. While I am disappointed with 24 pages I am more dissatisfied with the contents of the magazine.

I have been a subscriber since late 1983 and thank you and your predecessors for the loyalty and support shown for the TI99/4A computer. I learned 95 percent of what I know about computers and their language by sitting up to wee hours of the morning trying to find my mistakes (in typing) and attempting to make a program work. I would not be where I am today if it were not for MICROpendium.

A few years ago the magazine ran a "poll." The results showed the majority of users had a P-box, console with Extended BASIC, monitor, 32K memory and a pinter. Many had an RS232 board. A *minority* had other equipment. The many languages of the computer tend to confuse users and the greater part of the users agreed that Extended BASIC was the choice of use. Regina was challenged, by me, to convert to Extended BASIC, and even the magazine defended the cause of the many users who dealt in BASIC only.

Today the magazine prints programs that do not work. Lines are missing. Words are printed incorrectly (this month on Page 16 top line — CALL LINK("CUR SOR"). That item is what triggered me off and caused me to write this letter. Can you come close to understanding the frustrations the separation in CUR SOR caused?

I have *never* questioned how much a subscription cost for the magazine. I do question quality, or the lack thereof. If you are going to print a magazine for the TI99/4A computer, please print one that appeals to the membership. I and man others do not have the money for a Geneve, an Asgard Memory System, a CD-ROM, a SCSI, etc. I am sure you get the idea. And — the history of the TI99/4A and this month the complete background of the Apple computer did not excite me in the least.

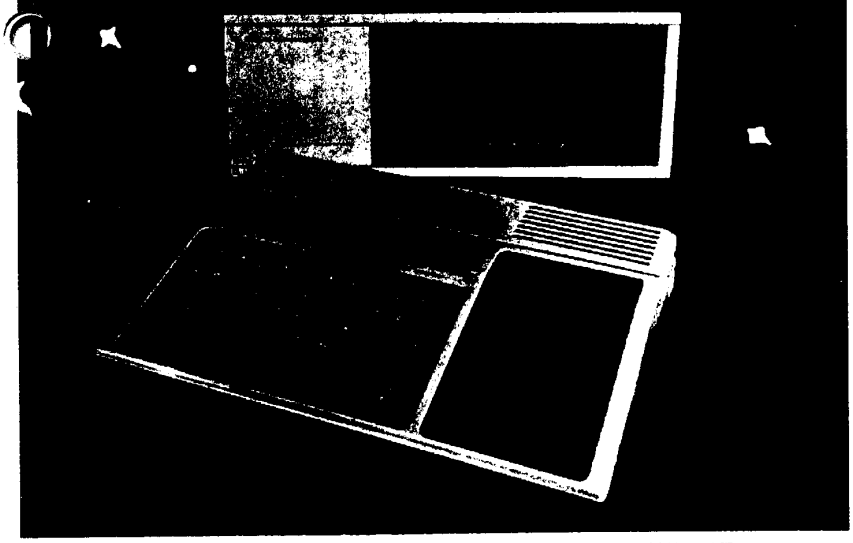
If you have read this far, thank you. My subscription runs till March 1996. If the quality of the magazine continues to deteriorate I will not be renewing. I appreciate that the lack of my subscription will not affect the MICROpendium balance sheet, but I hope my letter will cause you to scratch your head and wonder, "What is he trying to tell us?"

Harry Allston
Reedley, California

Thanks for sharing your thoughts. I won't go into the technical details as to why CURSOR turned into CUR SOR, suffice that it has to do with formatting programs for use in the magazine. It's a symptom of a larger problem that isn't likely to be resolved to anyone's satisfaction.

(See Page 6)

TI-99/4A™ CD ROM



IMAGINE 650 MEGABYTES OF SOFTWARE WRITTEN FOR THE TI99/4A AND 9640 ON ONE CDROM . THAT IS WHAT YOU WILL HAVE AT THE END OF YOUR SUBSCRIPTION TO THE TI 99/4A CDROM. FOR THE PRICE OF 30 -50 PUBLIC DOMAIN DISKS YOU WILL RECEIVE 1900 DISKS INITIALY AND MANY MORE IN PERIODIC UPDATES UNTIL YOU WILL HAVE 650 MEGABYTES OF TI SOFTWARE. THATS A LOT OF SOFTWARE FOR YOUR COST OF ONLY $100 . SEND YOUR ORDER FORM WITH A CHECK OR MONEY ORDER PAYABLE TO MILLBRAE CD ROM,350 MARCELLA WAY, MILLBRAE, CA 94030. VOICE:(415) 697-1108 ,FAX:(415)697-7406

YOUR NAME:_____

ADDRESS:_____

CDROM PLAYER AND PLATFORM INFORMATION:

MY CDROM PLAYER IS MADE BY:_____MODEL#:_____

I HAVE IT CONNECTED TO: IBM/COMPATIBLE___MAC___HORIZON SCSI___

IF YOU CHECKED HORIZON SCSI: IT IS CONNECTED TO:TI99/4A___9640___

I USE THE FOLLOWING OPERATING SYSTEM WITH MY CDROM PLAYER:

WINDOWS___DOS___VERSION:_____(3.1 FOR WINDOWS FOR EXAMPLE)

SYSTEM 7.X___MYARC DOS(MDOS)___VERSION_____ TI99/4A____

PLEASE INCLUDE YOUR CDROM PLAYERS SPEC SHEET WITH YOUR ORDER

TI CD ROM UPDATE :

WAITING FOR THE HORIZON SCSI CARD WE ORDERED IN APRIL!

DOES ANYONE WANT TO SELL US THEIRS?

OUR CD ROM SOFTWARE NEEDS THE TI EPROM TO BE FINISHED ON THE SCSI CARD AND BOTH TI AND GENEVE SCSI OWNERS WILL NEED THE SCSI CARDS CD ROM FIRMWARE ROUTINES PLUS ANY NEEDED MDOS EXTENTIONS FINISHED BEFORE WE CAN SELL THEM ANY CD ROMS

WHAT CAN WE SELL?

WE HAVE BEEN WORKING ON A BBS COMPATIBLE CD ROM AND EXPECT TO HAVE A BETA TEST VERSION BY SEPTEMBER IF NOT SOONER.

IBM COMPATIBLE OWNERS WILL BE ABLE TO USE THIS CD ROM ALSO TO DOWNLOAD TO THEIR TI OR 9640 USING X MODEM's BINARY TRANSFER AND TELCO AS ONE POSSIBLE SETUP. SUCCESSFUL DOWNLOADS USING A VERSION OF OUR BBS COMPATIBLE CD ROM WERE MADE AUGUST 5th.

IN ADDITION A TIEMULATOR VERSION BASED ON TIEMUL 5.1 IS CLOSE TOO WE HAVE NOT FOUND A WAY AROUND PC99's WASTEFUL FILE STRUCTURE WHICH WOULD REQUIRE 2 OR 3 CD ROMS TO HOLD THE SAME AMOUNT OF FILES AS A TI EMULATOR-COMPATIBLE CD ROM.

CHANGES!!! A DECISION TO INCLUDE AS MANY MYARC 9640 FILES AS WE COULD HAS MADE THIS CD ROM A TRUE REFLECTION OF THE TI COMMUNITY. IN ADDITION AND PARTLY BECAUSE OF THE EXPANDED SCOPE OF THE TI CD ROM IT WILL CONTAIN ARCHIVED FILES .

ALL CD ROM PLAYERS CAN READ OUR CD ROM NOW BUT YOU SHOULD STILL BUY A CD ROM PLAYER WHICH CAN BE USED BY BOTH YOUR COMPUTERS AND ONE WHICH SUPPORTS SOFTWARE WRITTEN FOR YOUR NON TI. THIS MEANS WE ARE STILL THINK YOU SHOULD BUY A SCSI DOUBLE SPEED CD ROM PLAYER NOT A SINGLE SPEED AND DON'T WASTE MONEY ON ANYTHING FASTER.

FEEDBACK

(Continued from Page 4)

It used to be that we worked full-time for MICROpendium and could put in the hours it took to produce it. Now, we both work full-time elsewhere and work on MICROpendium evenings and weekends. It's not quite a voluntary effort, but it's nearly that. The only reason we're continuing MICROpendium is because there is a loyal core of readers who want to read it. When that core starts to diminish, MICROpendium will disappear.

I realize that your comments are shared by others, and I have no problem with that. We work under severe limitations, both financial and in terms of time. We do the best we can afford to do. We'd like to do more, but we can't.

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

Author says RXB loader fulfills promise

I promised people that RXB could create a XB program from Extended BASIC and save that XB program to disk then run the very XB program it had just created. A bonus to that promise is this program deletes itself and the USER file it used to create the LOAD program then replaces the LOAD program loader that created both with LOAD.

So now that you are quite thoroughly confused as to what it does, let me explain why it does all this. What is needed in XB in my opinion is a decent loader from the XB environment. At present, Funnelweb or Menu or Boot and many other loaders do not run XB correctly because the loader they use messes up the XB random number seed, and even locks up the system, because of being Interrupt driven. I've done much needed research to keep absolute compatibility with all BASIC and XB concepts. Thus, I have never found a XB program that does not run correctly with no problems in RXB; also, I should mention if there was any problem it meant just having to delete several lines and substituting one line of RXB. Therefore, any programs with problems now only get smaller and faster.

Okay, all bragging aside let us get back to the promise I made. Original LOAD:

Line#	What it does.
100	Remind you what this program does.
110	Makes sure RXB version is higher then 600 or errors out.
120	Which drive was last used.
130	Get the name of the drive from the last card used.
140	Delete this program and set up for filename storage.
150 - 190	Types of files in variable TYPE$ (five types)
200 - 310	Catalog the disk with a standard subroutine. Thanks MicroPendium!
320	Is this a DV254 XB format? If yes save it, if not ignore it.
330	Is this a PROGRAM format? If yes goto 340 and if not goto 350
340	Ask if a XB or EA program image file. If XB or EA change T for type and if Space Bar just ignore file.
350	Is this a DF80 object file format? If yes goto 360 if not ignore.
360	Ask if a EA3 file. If yes goto 370 if not ignore.
370	X is file counter. F$(X) saves the names and type.
380	Single key stroke scan.
390 - 650	Explain what is happening and create the USER file.

This is really complicated to explain, so see the program LOAD it made. It also simplifies my having to explain all the quote marks needed.

660	Closes all open files. (#1 and #2)
670	Now use the USER file just created. This also ends this program.

Now to explain the program it creates. The user file first clears memory with NEW and starts with:

Line#	What it does
100 - 130	Lines are same as previously in last program but _$ is now named BT$.
140	Loop by reading data statements until a empty string is found. N is the counter, N$(N) stores the filenames, and T(N) stores the types of programs.
150	Show which drive was booted.
160	Turn off Quit Key. See if secondary counter C is larger then N.
170	Put letter and filename on left of screen. Check for empty string.
180	Put letter and filename on right of screen. Check for empty string.
190	End loop or continue looping to line 170
200	Indicate key choices.
210	Scan keys for letter to run program or next list of programs to run. If a letter is selected R$ has the entire drive and file name. If Space Bar then restart with rest of filenames goto 150
220	K is file type to run. 1 is EA3, 5 is EA5 and 4 is XB. 1001 - ???? are the data statements of filename and type. The last is empty.

The LOAD creator is 14 sectors and result LOAD is 5 sectors. RXB is required so use your GRAM device, Geneve or call Horizon to order a cartridge. Phone me at (360) 737-7963 or write.

Rich Gilbertson
1901 H St.
Vancouver, WA. 98663

1980's home computer era — Part 3

Atari: Bouncing off from Pong success

By BILL GASKILL

Last month's article covered Apple Computer and the machines they built during the late 1970s and throughout the 1980s. This month we cover the Atari Computer Corporation. Before getting into the meat of the material, though, you might enjoy knowing that the name Atari comes from the Japanese game of Go, which was a favorite of company founder Nolan Bushnell. Also, I am forever indebted to my good friend Steve Mehr of Thousand Oaks, California, for loaning me his copy of the book *Zap! The Rise and Fall of Atari*. It was a major source of information for this installment, largely on the early days of Atari. The book was published in 1984 and I have not been able to find a copy anywhere, so it is apparently out of print. If any reader has a copy for sale, I'd be interested in hearing from you. Anyway, thanks, Steve!

ATARI:

Atari is still in business today, but I haven't been able to discover what they produce or sell, other than some video games. I certainly haven't found what has sustained them financially over the last 6-8 years? I recently watched a PBS documentary about the Silicon Valley, done by a local historian. In the show this person did a small video byte of Atari headquarters and mentioned that they are still looking for their next "Pong" type hit, but he didn't tell the viewer what is sustaining the company today.

Just before Thanksgiving '94 (which is when I wrote this installment, by the way), I did find out that Atari has released a new 64-bit video game machine called the Jaguar. It is supposed to be the "ultimate" in realism with 3-D video and CD quality sound. This comes from the folks at Toys 'R Us, not Atari, so maybe it's actually true? Problem is, Nintendo is now (November 1994) saying that the video games market is going to be flat this Christmas (December 1994) because there is nothing new to sell. They have announced the impending release of a new video game player called Virtual Boy, due out in February 1995. Virtual Boy will simulate virtual reality (what Jeff Fahey and Pierce Brosnan experienced in the movie Lawnmower Man) by allowing the player to put their head up against a rest on the machine which blocks out all light, making the player more of a captive audience. Anyway, for Atari's economic well-being, I hope that the folks at Nintendo are wrong in their projections for a poor Christmas buying season. Atari is no doubt counting on the Jaguar to be a big seller. The machine I saw in Toys 'R Us was impressive looking, but it also was priced at $249.95, which is more than $100 higher than the best Nintendo or Sega system. The least expensive of the (only) six cartridges available for Jaguar was $49.95, with $59.95 seeming to be the average. Still, if the quality is all that it is hyped to be, maybe it will be a success. Recent sales history does not tell us that however. Witness the failure of the 3DO and Phillips CDI machines to crack the market. The game playing public seems to be more into quantity than quality. Guess we'll have to wait and see.

Atari, as you may already know, started life in 1972 when company founder Nolan Bushnell, a 29-year-old native of Clearfield, Utah, founded the company with the express purpose of inventing the first successful commercial video game. Pong was the outcome, followed four years later by the Atari 2600 VCS (Video Computer System) which is what gave the company the tremendous financial potential it had when Warner Communications bought Bushnell out for $28 million in 1976.

The VCS was conceived by Atari engineer Joe Decure who designed the chip set and the first prototypes; Harold Lee, who is credited with having pushed Nolan Bushnell in the direction of consumer electronics with Home Pong, and Steve Meyer, who also figured out how to make the VCS cost-effective. Although the VCS was designed by Atari under Bushnell, the company lacked the capital to produce it, which is where Warner entered the picture. They had the money and Atari had the product, so Warner bought Atari for $28 million, with Bushnell receiving $15 million for his own pockets. Part of the sale stipulated that Bushnell refrain from entering a competing business for seven years and that he stay on at Atari to see the VCS development materialize. He honored both contract stipulations and by November 1977 saw Atari produce 400,000 VCS units. He was paid a cool $100,000 a year for what he considered was "doing nothing," but hey, what the heck?

Going into 1977 the video game industry was packed with competitors.

— RCA came out with the Studio 2 video game system,

— Magnavox introduced Odyssey 2,

— National Semiconductor had the Adversary video game system,

— Fairchild Camera and Instruments, now called Fairchild Semiconductor, was releasing a new game a month for their Channel F product, (VAST 99er Ralph Rees has one of these if you've never seen one), and

— Allied Leisure was developing a new video backgammon game which they touted as being the next generation of electronic games.

Unfortunately for the folks at Warner Communications the video game market hit a severe slump in 1977, so Atari ended up sitting on an inventory of VCS units that no one was buying. The slump, however, produced a pretty sizable "shakeout" of the players in the video game industry that only Atari and Coleco would survive. When the market returned again in 1979, bigger than ever, due almost totally to the introduction of a game written in Japan called Space Invaders, Atari was perfectly positioned with a waiting inventory of

(See Page 8)

HOME COMPUTER ERA —

(Continued from Page 7)

new units to sell to a hungry game-playing public.

The resurgence in the video game market and the lucrative returns it yielded also spawned competition for Atari on their own machine. In 1979 former Atari software engineers Alan Miller, David Crane, Bob Whitehead and Larry Kaplan formed Activision and began producing their own game cartridges that would run on the Atari VCS. At the time, two-thirds of the game playing consoles in American homes were Atari systems and the average household was buying five cartridges a year.

No one had the guts to make software that was compatible with someone else's machine until Activision did, but the fact that they sold $65.9 million in software in 1980, only a year after the company was formed, certainly gave credence to the idea.

Impressed by the Activision success, another former Atari employee, marketing VP William Grubb, formed Imagic in 1981. Atari software engineer Rob Fulop jumped ship and joined Imagic in 1982 where he wrote Demon Attack. That one program generated $30 million in sales for Imagic in their first year of business.

In 1980 and 1981 Atari was the darling of the stock market and, of course the video game industry. We've all heard about TV's Emmy Awards, the film industry's Oscar, and Broadway's Tony Award, but did you ever hear of the Arkie? *Video Magazine* created the Arcade Awards in 1980 to honor outstanding achievement in the field of electronic gaming. Known affectionately as the Arkies, by 1982 the annual awards were jointly sponsored by *Video Magazine* and *Electronic Games* magazine. In the first three years that the awards were given, Atari walked away with eight winners in the home video category plus some in the commercial arcade category.

In 1980 Atari won Arkies for Basketball, Video Olympics, and Air-Sea Battle.

In 1981 Atari won Arkies for Adventure and Superman.

In 1982 Atari won Arkies for Asteroids, Missile Command and Warlords.

Some of the reasons or innovations the awards were given for are very interesting in that they show some groundbreaking ideas developed during the infancy of the personal computer, home computer and home video game industry, that we would laugh at today. What was so new and so innovative in 1980 is so taken for granted today that it is not even noticed by the computer/game-using public.

In 1980 the judges praised Basketball's use of a trapezoidal court to simulate depth of field (three-dimensional effect). In 1981 fantasy gaming was successfully translated to the video game screen for the first time with Adventure. In 1982 Missile Command won special praise for allowing a choice of starting levels, so the experienced players could skip over the novice attack waves.

By 1982 VCS sales began to falter, as did sales of Atari-produced video games. Third-party cartridge manufacturers were producing better, more sophisticated software (like Pitfall and Freeway) and both Mattel and Coleco had introduced more powerful machines when they brought out the Intellivision and ColecoVision products. Atari countered with the 5200 that was introduced at the 1982 Summer Consumer Electronics Show, but they made the fatal mistake of introducing a machine that was not compatible with existing 2600 VCS software. An adapter was quickly announced when they realized the mistake, but it didn't ship with the new 5200, so consumers rejected the 5200 en masse. Consequently, the 5200 was never the success Atari hoped it would be, and it was never able to help Atari overcome the loss of market share that had been given up to Mattel and Coleco.

The following are excerpts from an article entitled "Sneak Peeks" that appeared in the September/October 1982 issue of *Atari Age* magazine.

"Samples of the Atari 5200, the new advanced home video game system due in October, were up and running at the Consumer Electronics Show, and the crowds waited patiently to try their hand at superbly detailed versions of Soccer, Galaxians, Missile Command and Space Invaders.

"The biggest news at CES was not the specific games being offered, though, but two surprise additions to the game system itself — a Trak-Ball controller, and an adapter to make current VCS cartridges compatible with the 5200 system.

"We were especially happy to learn about the adapter unit for the Atari 5200, which lets VCS owners play the game cartridges they already own through the new system. While gameplay for VCS cartridges will remain the same when played through the 5200, the adapter lets you enjoy your current cartridge collection and the exciting new games programmed exclusively for the new 5200 without having to switch game consoles. The adapter is due out in 1983."

Also in 1982, sales of Atari's 400 and 800 home computers, which they had introduced in 1979, began to take a battering at the hands of Jack Tramiel and Commodore. Sales of the very popular VIC-20, which had been around since June 1980, and the newly introduced Commodore 64, were taking their toll on Atari's line of home computer. By September 1983 Atari had lost half a billion dollars between VCS software, the VCS, the 5200 and the 400/800 home computers (so TI wasn't the only victim of a Jack Attack). As a result of this financial debacle Warner Communications ousted Raymond Kassar as company CEO in September 1983 and installed James Morgan, a former Philip Morris executive, as Atari CEO.

On top of layoffs that had already taken place, Morgan laid off 250 more Atari employees, canceled the introduction of the powerful 1400XL computer and worked hard to reestablish Atari's vision of who it was, and where it was going. He hired Alan Alda to help sell Atari products just as TI had chosen Bill Cosby, and Commodore had tapped William Shatner to push their goods. But before Morgan could really get started, Jack Tramiel entered the picture in July 1984 and bought Atari right out from under any plans Morgan had for Atari's future.

Columnist John J. Anderson, writing in the October 1984 issue of *Creative Computing* magazine, gives us his thoughts on how Atari went from being the premier game machine company in the world in 1982 to a firm that had lost $538.6 million by 1984.

(See Page 9)

HOME COMPUTER ERA —

(Continued from Page 8)

"...I can't resist getting in a quick 'I told them so.' If only they had made the 5200 game machine 400/800 compatible and offered an optional keyboard peripheral. If only they had killed the 1200 on the drawing board. If only they had brought out the 1450 last fall. If only they had acted early to change their image. If only they had protected the morale and egos of their most creative minds. If only they had realized that the videogame and the low-end home computer were no longer separate markets. If only they had cut costs without cutting quality. If only the XL series had been truly compatible with the old Ataris."

Michael Tomczyk, author of *The Home Computer Wars*, provides us with another view of Atari, in the days immediately following Jack Tramiel moving in. Here are some excerpts from his book.

"A shell-shocked Atari employee told me that on one of those first days in the company, Jack was being shown some new products, and to show what he thought of them he dumped them on the floor."

"In less than a month the world-wide Atari staff was reduced from 5000 to less than 1500."

"Atari was bloated with physical facilities as well as people. On the day Jack (Tramiel) and his team started, Atari occupied over 40 buildings, most of them leased. By the end of the first week the total was down to 7 buildings."

Between 1984 and 1987 some drastic changes took place at Atari, most significantly, of course, the sale of the company to the very person responsible for bringing Atari to its knees. The deal was that Tramiel would buy Atari from Warner Communications for $240 million, but that in essence Warner would loan Jack the money to buy it, and at below market interest rates.

As the new owner of Atari one of Tramiel's first acts was to oust Morgan and place his sons Sam, Gary and Leonard in key positions. Sam became company president, Gary was given the job of collecting Atari's $300 million in accounts receivable and Leonard was placed in charge of software. Once this was accomplished, Tramiel concentrated on the introduction of the Atari ST line of computers. The STs were designed by Atari (Tramiel) to compete directly with the Amiga Lorraine that Commodore International had purchased from Amiga company founders Jay Miner and David Morse in 1984.

Before the introduction of the ST line, media moguls like John J. Anderson were singing the praises of Tramiel:

"Nobody is really sure what Jack will do to and for the Atari product line. It seems likely that the Atari 800XL will continue to be sold, at least through early 1985. As for anything else, all bets are off. It is now highly unlikely that the 1450XL, with built-in parallel bus disk drive and modem, will ever see the light of day. Jack is savvy enough to know that the 1450 is last year's product. He want's to get next year's product out the door as soon as possible, and the 1450XL is not it."

Under Tramiel, Atari's new motto became *Power Without the Price*. Six new computers appeared at the Winter 1985 Consumer Electronics Show, only five months after Tramiel bought the company. Four of the new machines were eight-bit powered computers and two were the new ST line, nicknamed the "Jackintosh" machines, because of their Motorola 68000 CPU and GEM (Apple Macintosh-like) operating environment.

Atari shocked the computer world with pricing never before heard of in computers and peripherals. For example, a 15mb hard disk for the new Atari 130ST/520ST machines retailed for $399! This was an incredible price breakthrough for 1985 when a 10mb hard disk was typically still selling in the $1000 range. You may recall that the Atari 1040ST was also the first computer to offer more than a megabyte of RAM for under $1000.

Apparently none of Atari's pricing strategies worked. While the ST line did become reality, it apparently never even came close to being the savior for Atari that Tramiel had hoped it would become. For reasons I have not yet discovered, like maybe Atari was never able to deliver on all of its fantastic promises, in a matter of three years the ST was in trouble, despite all of Tramiel's efforts to promote it. Following is a comment taken from the September 1988 issue of *Compute! Magazine*, made by contributing editor Arlan Levitan:

"Software development for the Atari ST line is in the dumper, with most software houses blaming poor sales on the current slowdown in ST sales and a higher than normal amount of software piracy in the ST market."

Early 1985 projections had Atari selling hundreds of thousands of STs, but when it came time to make a public offering of the "new" Atari stock, the numbers showed only 150,000 units shipped. Things got worse as the weight of the PC Clone explosion pressed harder and harder on the GEM-based Atari STs until the entire ST line seemed to fade into oblivion by 1990.

In January 1986 Atari introduced a smaller, sleeker and lighter 2600 video game machine (the VCS was dropped from the new machine's name) that hits the streets with a suggested retail price of $49.95. Though the machine was nine years old by that time video games and video game machines were selling well enough so that Atari also introduced the 7800, which was a machine the pre-Tramiel Atari had designed as a replacement for the 5200 back in 1984. The 7800 hits the streets with a suggested retail price of $79.95 and it came with a new version of Pole Position, the popular road racing game.

The 7800 had been sitting around gathering dust when it was brought back to life by the Tramiels. Thanks to a chip named MARIA, the 7800's graphics were superior to even the eight-bit machines of the day and the 7800 accepted 2600 VCS cartridges without having to use an adapter. It competed right along side Nintendo and then later Nintendo and Sega machines in the Toys 'R Us stores around the country for quite a few years. The 7800 finally succumbed to the superior software that was being released for the Nintendo and Sega machines and kids just stopped buying Atari stuff.

Somehow, despite all of its trials and tribulations, Atari remains in business today, with the tiny Atari Portfolio handheld computer being the only new unit it has introduced in more than four years (not counting the new Jaguar). Although I nev-

(See Page 10)

HOME COMPUTER ERA —

(Continued from Page 9)

er owned an Atari home computer, my wife Jacque and I were right in there with the first buyers of the 2600 VCS in the late '70s. In fact, she was the one who discovered video games, not me. We also still have the 2600 VCS machine and some 35 cartridges for it set up today, ready to battle each other in a game of Breakout at the first sign of a rainy day.

Following is a list of Atari's contributions to the "home computer" market of the 1980s. Some, like the 1400XL and 1450XLD were announced and even shown at CES, but were never released.

• **ATARI 65XE** — Basically a repackaged 800XL that retailed for $99 with 64K RAM, built-in BASIC, 256 colors, a four-voice sound chip, 11 graphics modes, five text modes, sprites, what Atari called player/missile graphics, an international character set, a cartridge slot, a serial bus for peripherals, and two joystick ports.

• **ATARI 65XEM** — This was a 65XE with an "Amy," which was an eight-voice sound chip. According to the editorial staff of *Compute!* (page 8, April 1987) it was never actually released, but was set to sell for $150 retail.

• **ATARI 65XEP** — A portable 65XE, Atari's version of the Commodore SX-64, came with a rechargeable battery pack good for three hours and also had a 3.5-inch floppy drive, which was a relatively "new" idea in 1985. It sold for $399.

• **ATARI 130XE** — This computer was basically a 65XE with 128K RAM and a rear parallel connector. It retailed for $150 but was available for $129 on the street.

• **ATARI 400/800** — The 400 and 800 were the original home computers produced by Atari in 1979. The 400 had 8K RAM and was not expandable. The 800 was expandable up to 48K RAM, had a full keyboard, two slots for cartridges, and custom sound and graphics chips designed by Jay Miner, who would later design the Amiga computer.

• **ATARI 600XL, 800XL, 1200XL, 1400XL, and 1450XLD** — These were the flagship line of home computers during the Warner Communications era of ownership. As I stated previously, though, the 1400 and 1450 were never actually released. The 1200XL was considered the top of the Atari computer line, but it was unpopular with consumers so it lasted only a year. The 600XL then became the bottom end of Atari's eight-bit line and the 800XL the top of the eight-bit line. By mid-1987 the 600XL would be gone, but the 800XL was still available for $69.99 on the street.

• **ATARI 130ST, 520ST and 1040ST** — The GEM-Based STs proved to be quite popular in Europe, but not in the U.S. They were all basically the same computer except for RAM. Each sported 192K ROM expandable to 320K with a plug-in cartridge, 512 colors, parallel and serial ports, floppy and hard disk interfaces, MIDI interface two Atari joystick ports, TV or RGB ports, three-voice sound synthesizer, 94-key keyboards with numeric keypads, 10 special function keys, TOS (Tramiel Operating System) and the Graphics Environment Manager (GEM). The 130 sold for $399, the 520 for $599 and the 1040 for $999. The 1040 had an 8mhz clock speed.

Although they were announced at the January 1985 Consumer Electronics Show, a year later they were still not readily available, evidenced by the following comments made by Glenn Hartwig, *Byte*'s technical editor of reviews, in the November 1985 issue, page 253.

"Another relatively new arrival here (at *Byte*) is the Atari 520ST, and since a closer look (eventually a full review) is under development, I'm not going to anticipate the reviewer's comments to any significant degree. The almost total lack of applications software at this point is an obvious drawback, but we'll withhold judgment on that front until and unless Atari and software developers start to show their wares. At the very least, , however, putting almost completely naked hardware out in front of the public would seem to indicate that the company has a fairly high degree of faith in its users' curiosity and enthusiasm."

In the January 1986 *Byte*, you will find an Atari 520ST product description, not a review, which they did for the following reasons.

"Some of the equipment we received such as the hard-disk drive, were prototypes, and at the time of this writing *(which was probably October 1985)*, software is scarce. Atari has not yet completed its BASIC interpreter, and the operating system, TOS, remains unfinished."

• **ATARI MEGA STs** — These were Atari ST machines with 4mb of RAM. They were announced at the January 1987 Consumer Electronics Show.

• **ATARI 1040STE** — Announced at the Winter Consumer Electronics Show in Las Vegas, Nevada, on January 14, 1991. It was designed specifically for MIDI uses by musicians and was touted as being so powerful it would even desktop publish a musical score. I'm not able to verify that any were actually ever produced however.

• **ATARI PCs** — In January 1987 Atari, like everyone else in the computer world, was "Turning Blue," meaning they were jumping on the IBM (Big Blue) PC Compatible bandwagon by producing a computer(s) that would operate in the MS/PC DOS environment. One model sold for $499, and the other for $699. They came with DOS 3.2, GW-BASIC and the GEM operating environment.

"Both of the Atari PCs are aimed at the burgeoning clone market now ruled by Tandy, Leading Edge, Epson and Blue Chip...Both models include 512K RAM (expandable to 640K on the motherboard); and internal 5.25-inch floppy disk drive, RS-232 serial and Centronics-standard parallel ports, a mouse port and mouse controller; composite and PC standard RGBI video outputs; software switchable clock speeds of 4.77 megahertz (the same as the IBM PC) and 8 megahertz (turbo mode); a socket for an 8087 math coprocessor; a PC-style detached keyboard; and a built-in color graphics adapter."

While I don't ever remember seeing any in the stores in my town, they probably were on somebody's shelf somewhere in the U.S.? I clearly remember seeing the Commodore Colt show up at Wal-Mart, but I can't recall ever seeing an Atari PC except in magazine photos. The point being, I can't personally verify whether either computer actually made it beyond being announced at CES in January 1987 and really did find a place on some retailer's shelf.

(See Page 11)

HOME COMPUTER ERA —

(Continued from Page 10)

• **ATARI GAME MACHINES** — We all know about the VCS 2600 system that Nolan Bushnell's crew invented, because it had such an impact upon American homes. But there were also the 5200 and 7800 ProSystem game players that came in 1983 and 1984. Although both machines sported graphics far superior to the 2600, neither ever reached its popularity.

• **JAGUAR** — A 64-bit (it's really two 32-bit processors), round, black plastic console with curved cartridges that plug into the top of the machine. It is stylish and expensive at $249.95, but it may turn out to be the next level of video game sophistication? Atari advertises it as having 3D graphics and CD quality sound. Introduced in November 1994.

• **LYNX** — A handheld, color video game machine that didn't catch on as well as it should have. In 1991 Lynx sold for right around $149 with ComLynx cables, an AC adapter and the California Games cartridge, or $99.95 for the basic unit.

• **MY FIRST COMPUTER** — This was not a computer, but a rather intelligent computer keyboard that Atari designed in 1983 as an add-on to the more than 10 million VCS 2600 units it had sold since 1977. The purpose was to give Atari another shot at reaching the huge installed base of VCS owners, and it would provide VCS owners with a way to make their game machine into a "real computer". The peripheral was to retail for just under $90, it was powered by a 6502 CPU, it would include 8K of RAM expandable to 32K, upper and lower case character sets, support for 16 colors and support for a 32-column by 24-row display. MFC also had two sound generators, a cassette interface and an expansion slot for plugging in game cartridges, memory expanders and other peripherals. The peripherals, though, were to be ones that were manufactured just for the MFC because it was incompatible with all existing Atari add-ons for the "bigger" machines like the 600, 800 and 1200. Although MFC is featured in the May 1983 issue of *Compute!* on page 44, I never saw one in the flesh and wouldn't be surprised if Atari canceled plans to produce it before MFC ever hit the stores. It just didn't make that much sense in the continually eroding profit margins of the home computer market circa 1983.

• **PORTFOLIO** — The first "Palmtop" computer. It came with 128K RAM that was expandable to 640K. It had bundled software like Lotus 1-2-3 and a couple of other programs, as I recall. A neat idea that didn't seem to catch on. In usable configuration, it ran around $700.

TI-Chips to host 1996 MUG Conference

By GLENN BERNASEK

The TI-Chips, a TI-99/4A and Geneve user group in North Royalton, Ohio, will host the TI-99/4A / Geneve Multi-user Group Conference in the Cleveland, Ohio area in May or June of 1996. The date and place will be announced as-soon-as-possible.

The 1996 MUG in Cleveland is not a "take away" from Lima. But rather to give the Lima user group a much needed and well-deserved break. Charles Good, the internationally renown Lima User Group librarian, has expressed his desire for another user group or groups to host this popular affair and give his people a chance to attend a conference somewhere else for a change.

Good has also made it clear that it has been increasingly difficult to schedule the MUG conference on the Ohio State University-Lima campus. Therefore, it was clear that something should be done to keep this valuable and popular conference viable in Ohio.

Giving the Lima people a break and continuing the grand tradition established by Lima was reason enough for the TI-Chips to take up the reigns for 1996.

Good said he is thrilled to have another user group give him and the Lima users a break. He offered to assist the TI-Chips. He wants the 1996 MUG in Cleveland to be as every bit as successful as the MUGs in Lima have been for the last five years.

For more information or comments, via email, contact:
Glenn Bernasek, dd314@cleveland.freenet.edu
Chris Bodenmiller, by737@cleveland.freenet.edu
Jim Krych, ab453@cleveland.freenet.edu

THE ART OF ASSEMBLY — PART 50

Delays, again?

By BRUCE HARRISON

Remember last month? We said something about programmers being likely to redesign programs over and over. True to our word, we've made still more variations on the "delay" theme, so this month we'll pick up where we left off. This month's sidebar has, among other things, another complete program that doesn't do much, but serves to illustrate a couple of nifty ways of doing things.

DELAY WITH OPTION

The program DELAY4 uses a modified version of the delay subroutine used in DELAY3 last month. This time, the programmer using the subroutine has the option of allowing the delay to be terminated by a keystroke or not. A second word of DATA following the BLWP @DELAY instruction signals the "keystroke" option. As before, the first word of DATA tells DELAY how many 60ths of a second to run. The second word can be either zero or any non-zero number. If it's zero, then keystrokes during the delay period will have no effect. If it's non-zero, then any keystroke during the delay will cut the delay short. In the program shown, the first BLWP @DELAY has its second DATA word as zero, so while that message "DELAYING TEN SECONDS" is on the screen, pressing keys (other than Function-=) will have no effect, and the user will just have to wait for the 10 seconds to expire. When the second delay starts with the message "ANOTHER TEN SECONDS" on-screen, the "keystroke" abort is active, so pressing a key will terminate the delay early. Of course, if no key is pressed, the delay will continue until the full 10 seconds has passed. That's so because the second DATA word on this BLWP @DELAY is non-zero.

ANOTHER POTENTIAL USE

After the second delay in this program, we use the delay routine in another way, to produce a "slow reveal" of a message on the screen. Here, we delay by one-tenth second after each character in the message, unless that character was a space, in which case we skip over the delay. This has a key data word of zero, so that keystrokes will have no effect during the slow printing of the "PRESS A KEY TO EXIT" message.

The source code in the sidebar is well annotated, so our loyal readers should have no trouble following its operations. Some tricks are employed, such as taking advantage of the fact that the messages are right after one another in memory, so we can A R2,R1 to move our VMBW pointer on to the next message.

We also took full advantage here of the lengths of the messages being close, so rather than LI R2 with the new lengths, we were able to just DEC R2, DECT R2 and INCT R2. That saves some memory, but of course it can only be used if the next message is only one or two characters different in length.

WHAT ABOUT EXTENDED BASIC?

Having gone to such lengths to provide delays for our assembly readers, we felt that we should do something for our friends who program in Extended BASIC. The common way of doing delays in XB is, of course, with a simple FOR-NEXT like this:

```
FOR DELAY=1 TO 500 :: NEXT DELAY
```

That can be used in a subroutine or even a subprogram, and a variable can be used in place of the 500 shown, so that the subroutine can produce different delays. This works okay so long as it's used on the same machine it was written on. If, however, one takes such a program to a Geneve, the delays will all be too short. Conversely, if the writer of the program used a Geneve or a Bus-modified TI, all the delays will be too long when it's run on a standard TI.

Once again, assembly language comes to the rescue. We've made a special version of the delay routine tailored for use with XB programs. This version is sort of a "do everything" routine, so that the XB programmer can choose which features get implemented by changing the way the CALL LINK is written. Let's start with the case where what's desired is a simple delay of 4.5 seconds. The linkage would look like this:

```
CALL LINK("DELAY",4.5)
```

This will cause an absolute delay of 4.5 seconds, regardless of the computer in use. Nothing except Function-= (QUIT) will abort the delay.

Now suppose the XB programmer wants to allow the user to terminate the delay by pressing any key. To do that, just add a second parameter to the call link:

```
CALL LINK("DELAY",4.5,1)
```

That second parameter can be any non-zero number, or even a variable. If the second parameter happens to be zero, the effect is the same as the first case, and pressing keys will have no effect.

Okay, let's carry this one step farther. Suppose you want the ability to have any keystroke abort the delay, and you also want your XB program to know which key was pressed. For that, you make sure the second parameter is non-zero, then add a variable name as the third parameter:

```
CALL LINK("DELAY",4.5,1,K)
```

In this case there are two possible outcomes. If the user does not press a key during the delay, then the delay will run its full time, and the variable K will be zero after the delay is finished. If a keystroke (other than Function-4 or Function-=) aborts the delay, the ASCII value of that key will be in the variable K. If, for example, the user terminated the delay by pressing A, then K will equal 65 upon exit from CALL LINK. If the user presses Function-6, K will equal 12, and so on.

Any of the three parameters may be a variable instead of a number, but the third one must be a variable if it's included. Range for the first parameter is from 1/60 through 546 seconds. The lowest number, 1/60th, will probably never be noticed. The highest number, 546, will result in a 9 minute and 6 second delay. Thus, we don't recommend pushing the limits on these parameters. For ordinary delays in the range of, say, 1 second through 30 seconds, no problems should be noticed. Whenever the second parameter is

(See Page 13)

THE ART OF ASSEMBLY —

(Continued from Page 12)

present and non-zero, Function-4 will be active to break the program during the delay, unless the program has executed ON BREAK NEXT before the CALL LINK. In that case, pressing Function-4 will abort the delay, and K will equal 2, which is the value of the Function-4 key.

Regardless of any parameters, the Function-= key press will perform its usual function during the delay, sending the computer back to startup conditions. This will not be affected by an ON BREAK NEXT statement in the program.

THE SOURCE CODE

Part Two of this month's sidebar has the source code for this little routine, which we call the "ultimate delay." This has the usual features you'd find in a routine written for XB use, including the EQUates at the beginning, the use of NUMREF to get parameters, and of NUMASG to assign the key's value if required. We've also used XMLLNK to perform multiplication of floating point numbers and to convert the answer to integers for our use. Notice that we find out the number of parameters present by moving the number from >8312 into R6, then use that to govern what the routine does. If you forget to supply any parameter, the routine will simply exit without any delay. So long as one parameter is provided, and is in the correct range, all should go well.

Once again, assembly language comes to the rescue. We've made a special version of the delay routine tailored for use with XB programs.

Assuming there's one parameter, the routine gets the first one with NUMREF. Since this number is a floating point quantity in seconds, and our counting will be done in 60ths, we place the floating point number 60 in the eight bytes starting at ARG (>835C), then use an XML service to multiply the number in seconds by 60. XML gets used again to convert that result (at FAC) to an integer word. We then move that word to R4.

Just after the MOV to R4, we check to see if the number in R4 is zero or negative, and exit if either is the case, since we needn't bother delaying by zero, and we can't delay by a negative amount of time. Given that we have an acceptable number in R4, we check R6 to see whether we need to fetch the second parameter. If there's only one, we clear R5 so we'll "know" not to look for keys being pressed.

If there's a second parameter, we fetch that with NUMREF. As always, NUMREF places that parameter as a floating point number at FAC. Since we don't care about this number except whether it's zero or non-zero, we take advantage of the fact that the F.P. representation of zero makes the word at FAC equal zero. Thus we just MOV @FAC,R5, and that gives us what we want without need to convert to an integer.

The actual delay starts at label DLY0, where we double the number in R4 and clear the word at >83D6. That word is the screen time-out counter, which gets incremented by two on each vertical interval. (That's not what the Editor/Assembler manual says, but it is what happens.) Now the delay loop starts at label DLY1. Each time through the loop, we allow interrupts briefly, so that the counter can work. We then check R5, to see whether we are supposed to check the keyboard.

If R5 is zero, we skip ahead to CKDLY. If R5 is not zero, we scan the keyboard. If a key has been pressed, we capture the key's value in R3, then jump ahead to label KEXIT, so we can report out the key value if necessary. If no key was pressed, we get to label CKDLY, where we compare the number in the time-out counter to our desired delay amount in R4. If those numbers are not equal, we jump back to DLY1 and repeat the process. If they are equal, we move on to label EXIT, which clears R3. That means that if the delay has run its full time, the key value will show up as zero.

Label KEXIT right-justifies the key value in R3, then checks the number of parameters in R6. If that's less than 3, it means we don't have to report the key value to XB, so we skip ahead. If R6 was 3 or more, we go ahead and put R3 at label FAC, set R1 to 3 for the third parameter, convert the number at FAC to floating point format, and assign that to the variable named as the third parameter.

Label NKEX starts the last part of the code. Here, we clear the screen time-out counter, then load the GPL Workspace, and branch to the GPL Interpreter at >6A. That puts XB back in control of the computer, and of course what happens next depends on what's in your program right after the CALL LINK statement.

As with so many other of these utility items, we've made this routine DELAY available as part of a public domain disk, and have sent a copy to our friend Dr. Charles Good at the Lima Users' Group. Thus, you or your group can easily obtain this routine, complete with its source code, instructions for using it and so on. That address is:

Dr. Charles Good, P.O. Box 647, Venedocia, OH 45894.

We've called this disk The Ultimate Delay, because we think it offers all the flexibility anyone could ask for, and because it will yield reasonably accurate delay timing regardless of whether it's used on a TI or Geneve. For European users, whose systems work at 50 Hz instead of 60 Hz, there's a European version on the same disk. (In this case, Europe includes the U.K. and Australia.)

Next month we promise there will be no more delays, but will try to surprise you with some other topic. See you then.

SIDEBAR 50

```
* SIDEBAR 50
* PART ONE - A COMPLETE E/A PROGRAM
*
* DELAY4/S
* A FOURTH DELAY METHOD
* WITHOUT USER INTERRUPT
* WITH OR WITHOUT KEYSTROKE TERMINATE
* FOLLOWING IS A COMPLETE PROGRAM
* THAT CAN BE ASSEMBLED & TESTED AS IS
* ILLUSTRATES USE FOR SLOW REVEAL
```

(See Page 14)

THE ART OF ASSEMBLY —

(Continued from Page 13)

```
* AND ORDINARY DELAY
*
*      Code by Bruce Harrison
*      27 November 1994
*      PUBLIC DOMAIN
*
       REF  VMBW,VSBW,KSCAN   REF UTILS
       DEF  START        DEFINE ENTRY
START  LWPI WS           LOAD WORKSPACE
       LI   R0,9*32+5    ROW 10, COL 6
       LI   R2,20        20 CHARACTERS
       LI   R1,DLYTXT    DELAY MESSAGE
       BLWP @VMBW        WRITE THAT
       BLWP @DELAY       USE VECTOR
       DATA 600          60THS TO DELAY
       DATA 0            NO KEY ABORT
       LI   R0,11*32+6   ROW 12, COL 8
       A    R2,R1        SECOND MESSAGE
       DEC  R2           19 CHARACTERS
       BLWP @VMBW        WRITE THAT
       BLWP @DELAY       USE DELAY
       DATA 600          60THS TO DELAY
       DATA 1            KEY ABORT ON
       LI   R0,13*32+7   ROW 14, COL 7
       A    R2,R1        "DELAY FINISHED"
       DECT R2           TWO SHORTER
       BLWP @VMBW        WRITE THAT
       AI   R0,63        DOWN 2 ROWS -1 COL
       A    R2,R1        "PRESS A KEY"
       INCT R2           TWO CHARS LONGER
       MOV  R1,R3        ADDRESS INTO R3
SLOWRV MOVB *R3+,R1       ONE BYTE TO R1
       BLWP @VSBW        WRITE THAT
       CB   R1,@ANYKEY   WAS IT A SPACE?
       JEQ  SLSK         IF SO, NO DELAY
       BLWP @DELAY       USE DELAY
       DATA 6            1/10 SECOND
       DATA 0            NO KEY ABORT
SLSK   INC  R0           NEXT SPOT ON SCREEN
       DEC  R2           DEC LENGTH
       JNE  SLOWRV       IF NOT ZERO, RPT
KEY    BLWP @KSCAN        SCAN KEYBOARD
       LIMI 2            INTERRUPTS ON
       LIMI 0            STOP THEM
       CB   @>837C,@ANYKEY KEY STRUCK?
       JNE  KEY          IF NOT, RPT
       LWPI >83E0        LOAD GPLWS
       B    @>6A         BACK TO E/A
*
* THE DELAY SUBROUTINE IS:
*
DELAY  DATA DLYWS,DLY0   BLWP VECTOR
DLY0   MOV  *R14+,R0      GET DESIRED DELAY
       SLA  R0,1         DOUBLE THE NUMBER
       MOV  *R14+,R1      GET KEY FLAG IN R1
       CLR  @>83D6       CLEAR TIMEOUT COUNTER
DLY1   LIMI 2            ALLOW INTERRUPTS
       LIMI 0            STOP THEM
       MOV  R1,R1        KEY TO STOP?
       JEQ  DLY2         IF NOT, JUMP
       BLWP @KSCAN       ELSE SCAN KYBRD
       CB   @>837C,@ANYKEY  KEY STRUCK?
       JEQ  DLYEX        IF SO, EXIT
DLY2   C    R0,@>83D6     COMPARE NUMBERS
       JNE  DLY1         IF NOT EQUAL, RPT
*      CLR  @>83D6       CLEAR TIMEOUT
DLYEX  RTWP              BACK TO CALLER
*
* DATA SECTION
*
WS     BSS  32           MAIN WORKSPACE
DLYWS  BSS  32           DELAY WORKSPACE
DLYTXT TEXT 'DELAYING TEN SECONDS'
SCND   TEXT 'ANOTHER TEN SECONDS'
DOVTXT TEXT 'DELAY IS FINISHED'
PAK    TEXT 'PRESS A KEY TO EXIT'
ANYKEY BYTE >20          KEYSTROKE COMPARISON BYTE
       END
*
* PART TWO - A COMPLETE SUBROUTINE
* FOR XB USERS TO MAKE DELAYS
*
* ULTDLY/S
* FOR PRECISE DELAYS FROM XB
* MAKES POSSIBLE DELAYS FROM
* 1/60TH SECOND THROUGH 546 SECONDS
* INVOKE FROM EXTENDED BASIC BY:
* CALL LINK("DELAY",T,U,K)
* WHERE T IS DESIRED DELAY IN SECONDS
*       U IS ZERO OR NON-ZERO
*       K IS VARIABLE TO REPORT KEYVAL
* U AND K MAY BE OMITTED FOR PLAIN DELAY
* K MAY BE OMITTED IF KEYSTROKE TO ABORT
* BUT NOT REPORTED BACK TO XB
*    CODE BY:  Bruce Harrison
*    PUBLIC DOMAIN
*    07 December 1994
*
       DEF  DELAY        DEFINE ENTRY POINT
*
* REQUIRED EQUATES
*
NUMASG EQU >2008          NUMERIC ASSIGN
NUMREF EQU >200C          NUMERIC REF
KSCAN  EQU >201C          KEYBOARD SCAN
XMLLNK EQU >2018          XML VECTOR
CFI    EQU >12B8          CONV F. P. TO INTEGER
CIF    EQU >20            CONV INTEGER TO F. P.
FAC    EQU >834A            FLOATING POINT ACCUM.
ARG    EQU >835C          FLOATING POINT ARGUMENT
STATUS EQU >837C          GPL STATUS BYTE
ARGNUM EQU >8312          NUM OF ARGS
FMUL   EQU >0E88          F. P. MULTIPLY
*
* MAIN CODE SECTION
*
DELAY  LWPI WS            LOAD OUR WORKSPACE
       MOVB @ARGNUM,R6    NUMBER OF ARGS TO R6
       SRL  R6,8          RIGHT JUSTIFY
       JEQ  EXIT          IF NO ARGUMENTS, EXIT
       LI   R1,1          FIRST PARAMETER
       BLWP @NUMREF       USE NUMREF
       LI   R9,SIXTY      POINT AT 60
       LI   R10,ARG       AND ARGUMENT
       LI   R4,8          EIGHT BYTES
MOVARG MOVB *R9+,*R10+    MOVE A BYTE
       DEC  R4            DEC COUNT
       JNE  MOVARG        IF NOT ZERO, RPT
       BLWP @XMLLNK       USE XML LINK
       DATA FMUL          MULTIPLY BY 60
       BLWP @XMLLNK       USE XML AGAIN
       DATA CFI           CONVERT TO INTEGER
       MOV  @FAC,R4       PUT DELAY # IN R4
       JEQ  EXIT          IF ZERO, EXIT
       JLT  EXIT          IF NEGATIVE, EXIT
       CI   R6,1          COMPARE ARGS TO 1
       JGT  GETTWO        IF GREATER, JUMP
       CLR  R5            ELSE CLEAR KEY FLAG
       JMP  DLY0          THEN JUMP AHEAD
```

(See Page 15)

Graphics printing

Extended BASIC programs show how to make it work

By RON WARFIELD

After all the playing with ART type programs, and printing of the pictures we created, I decided to examine the graphics printing from XBASIC.

I wrote three programs to show how the printer works.

To create a graphic to print, we have to draw what we want on graph paper. Since we are printing in 8-pin mode, we need graph paper eight lines high.

Row	Code
8	128 when this dot is full the number is 128
7	64 when this dot is full the number is 64
6	32 when this dot is full the number is 32
5^	6 when this dot is full the number is 16
4	8 when this dot is full the number is 8
3	4 when this dot is full the number is 4
2	2 when this dot is full the number is 2
1	1 when this dot is full the number is 1

Now when you draw your graphic onto the graph paper all you need to do is add up the numbers vertically, e.g., if the first column is all full the value will be 255.

When you use single-density graphics the dots on your graph paper have to be inside the vertical lines. If you use double-density mode you can have dots on the vertical lines. This chart (Fig. 1) is only an example of how the graphic is defined and is not the same graphic we use in the program.

The first program prints in single density.

PROGRAM DESCRIPTION

Line 100 clears the screen.

Line 110 opens printer and cancels linefeeds and carriage return.

Line 120 sets printer to 24/216th inch line spacing.

Line 121 sets single density graphics with 1 block and 224 dots.

Line 130 reads the graphic definition 24 times.

Line 140-143, character definition.

Line 170 tells program to read again until 24 is read.

Line 180 tells printer to do a CR and a LF.

Line 181 repeat of above, only for bottom half.

Line 190 closes printer.

(See Page 16)

Fig. 1

```
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|^|^|^|^|^|^|X|^|X|^|X|^^^128^^^|^|^|^|^|^|^|^|XX^|^|^|X|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|^|^|^|^|^|X|^|X|^|X|^|^^^^64^^^|^|^|^|^|^|^|XX^|XX^|XX^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|^|^|^|^|X|^|^|^|^|^|^|^^^^32^^^|^|^|^|^|^|XX^|^|^|XX^|^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|^|^|^|X|^|^|^|^|^|^|^|^^^^16^^^|^|^|^|^|XX^|^|^|^|^|^|^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|^|^|X|^|^|^|^|^|^|^|^|^^^^^8^^^|^|^|^|XX^|^|^|^|^|^|^|^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|¬|-|
|^|^|X|^|^|^|^|^|^|^|^|^|^^^^^4^^^|^|^|XX^|^|^|^|^|^|^|^|^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|^|X|^|^|^|^|^|^|^|^|^|^|^^^^^2^^^|^|XX^|^|^|^|^|^|^|^|^|^|
|-|-|-|-|-|-|-|-|-|-|-|-|———-|-|-|-|-|-|-|-|-|-|-|-|-|
|X|^|^|^|^|^|^|^|^|^|^|^|^^^^^1^^^|XX^|^|^|^|^|^|^|^|^|^|^|
     SINGLE DENSITY                         DOUBLE DENSITY
```

Add up each column for these values.

1 2 3 8 16 32 64 128 64 128 64 128 is the first graphic and

1 1 2 2 4 4 8 8 16 16 32 32 64 64 128 128 64 64 32 32 64 64 128 is the second

THE ART OF ASSEMBLY —

(Continued from Page 14)

```
GETTWO INC  R1            2ND PARAMETER
       BLWP @NUMREF       GET NUMBER
       MOV  @FAC,R5       MOVE TO R5
DLY0   SLA  R4,1          DOUBLE R4
       CLR  @>83D6        CLEAR TIMEOUT COUNT
DLY1   LIMI 2             ALLOW INTERRUPTS
       LIMI 0             STOP THEM
       MOV  R5,R5         CHECK R5
       JEQ  CKDLY         IF ZERO, SKIP
       BLWP @KSCAN        ELSE SCAN KEYBOARD
       CB   @ANYKEY,@STATUS KEY PRESSED?
       JNE  CKDLY         IF NOT, JUMP AHEAD
       MOVB @>8375,R3     GET KEYVAL IN R3
       JMP  KEXIT         THEN JUMP AHEAD
CKDLY  C    @>83D6,R4     COMPARE TO DESIRED NO.
       JNE  DLY1          IF NOT EQUAL, RPT
EXIT   CLR  R3            CLEAR REG 3 - NO KEY
```

```
KEXIT  SRL  R3,8          RIGHT JUSTIFY KEYVAL
       CI   R6,3          COMPARE ARGS TO 3
       JLT  NKEX          IF LESS, SKIP AHEAD
       MOV  R3,@FAC       MOVE KEYVAL TO FAC
       BLWP @XMLLNK       USE XML
       DATA CIF           CONVERT TO F.P.
       LI   R1,3          3RD PARAMETER
       BLWP @NUMASG       ASSIGN VALUE
NKEX   CLR  @>83D6        CLEAR TIMEOUT COUNTER
       LWPI >83E0         LOAD GPL WORKSPACE
       B    @>6A          GO TO GPL INTERPRETER
*
* DATA SECTION
*
WS     DATA 0             PRELOADED R0
       BSS  30            R1 THRU R15
SIXTY  BYTE 64,60,0,0,0,0,0,0  SIXTY AS F.P.
ANYKEY BYTE >20           COMPARISON BYTE
       END
```

GRAPHICS PRINTING —

(Continued from Page 15)

SINGLE DENSITY

```
100 CALL CLEAR
110 OPEN #1:"PIO.LF.CR"
120 PRINT #1:CHR$(27)&"3"&CH
R$(24)
121 PRINT #1:CHR$(27)&"K"&CH
R$(224)&CHR$(1)
130 FOR I=1 TO 24
140 PRINT #1:CHR$(1)&CHR$(2)
&CHR$(4)&CHR$(8)&CHR$(16)
141 PRINT #1:CHR$(32)&CHR$(6
4)&CHR$(128)&CHR$(64)&CHR$(1
28)
142 PRINT #1:CHR$(64)&CHR$(1
28)&CHR$(64)&CHR$(128)&CHR$(
64)
143 PRINT #1:CHR$(32)&CHR$(1
6)&CHR$(8)&CHR$(4)&CHR$(2)
170 NEXT I
180 PRINT #1:CHR$(13)&CHR$(1
0)
181 PRINT #1:CHR$(27)&"K"&CH
R$(224)&CHR$(1)
182 FOR I=1 TO 24
183 PRINT #1:CHR$(128)&CHR$(
64)&CHR$(32)&CHR$(16)&CHR$(8
)
184 PRINT #1:CHR$(4)&CHR$(2)
&CHR$(1)&CHR$(2)&CHR$(1)
185 PRINT #1:CHR$(2)&CHR$(1)
&CHR$(2)&CHR$(1)&CHR$(2)
186 PRINT #1:CHR$(4)&CHR$(8)
&CHR$(16)&CHR$(32)&CHR$(64)
187 NEXT I
189 PRINT #1:CHR$(13)&CHR$(1
0)
190 CLOSE #1
```

The second program prints double density.

This is the same as the first program except, to fill the same amount of space we need to double the graphics read. So line 130 and 182 are doubled. Also lines 121 and 181 are altered to print more columns. Remember that graphics are printed in blocks of 256 columns. In the first program we printed in single mode so the columns or dots per row is 480, so 20 characters times 24 times read equals 480. Now the formula is print 1 block of 256 and 224 is left over which adds up to 480. See lines 121 and 181 above.

Below in line 121 and 181 we set double density and print three blocks of 256 and 192 dots left over to add up to 960 dots.

Single density is 480 dots while double density is 960 dots.

You have to make sure the numbers and the columns are the same or the printer locks up or prints garbage.

DOUBLE-DENSITY

```
100 CALL CLEAR
110 OPEN #1:"PIO.LF.CR"
120 PRINT #1:CHR$(27)&"3"&CH
R$(24)
121 PRINT #1:CHR$(27)&"L"&CH
R$(192)&CHR$(3)
130 FOR I=1 TO 48
140 PRINT #1:CHR$(1)&CHR$(2)
&CHR$(4)&CHR$(8)&CHR$(16)
141 PRINT #1:CHR$(32)&CHR$(6
4)&CHR$(128)&CHR$(64)&CHR$(1
28)
142 PRINT #1:CHR$(64)&CHR$(1
28)&CHR$(64)&CHR$(128)&CHR$(
64)
143 PRINT #1:CHR$(32)&CHR$(1
6)&CHR$(8)&CHR$(4)&CHR$(2)
170 NEXT I
180 PRINT #1:CHR$(13)&CHR$(1
0)
181 PRINT #1:CHR$(27)&"L"&CH
R$(192)&CHR$(3)
182 FOR I=1 TO 48
183 PRINT #1:CHR$(128)&CHR$(
64)&CHR$(32)&CHR$(16)&CHR$(8
)
184 PRINT #1:CHR$(4)&CHR$(2)
&CHR$(1)&CHR$(2)&CHR$(1)
185 PRINT #1:CHR$(2)&CHR$(1)
&CHR$(2)&CHR$(1)&CHR$(2)
186 PRINT #1:CHR$(4)&CHR$(8)
&CHR$(16)&CHR$(32)&CHR$(64)
187 NEXT I
189 PRINT #1:CHR$(13)&CHR$(1
0)
190 CLOSE #1
```

The third program prints my name in double density.

MORE DOUBLE-DENSITY

```
100 CALL CLEAR
110 OPEN #1:"PIO.LF.CR"
120 PRINT #1:CHR$(27)&"3"&CH
R$(26)
121 PRINT #1:CHR$(27)&"L"&CH
R$(74)&CHR$(1)
130 FOR I=1 TO 10
140 PRINT #1:CHR$(255)&CHR$(
129)&CHR$(129)&CHR$(129)&CHR
$(129)&CHR$(129)&CHR$(129)&C
HR$(66)&CHR$(60)&CHR$(0)&CHR
$(0)
141 PRINT #1:CHR$(127)&CHR$(
128)&CHR$(128)&CHR$(128)&CHR
$(128)&CHR$(128)&CHR$(128)&C
HR$(128)&CHR$(127)&CHR$(0)&C
HR$(0)
142 PRINT #1:CHR$(255)&CHR$(
64)&CHR$(56)&CHR$(4)&CHR$(3)
&CHR$(0)&CHR$(0)&CHR$(0)&CHR
$(255)&CHR$(0)&CHR$(0)
170 NEXT I
180 PRINT #1:CHR$(13)&CHR$(1
0)
181 PRINT #1:CHR$(27)&"L"&CH
R$(74)&CHR$(1)
182 FOR I=1 TO 10
183 PRINT #1:CHR$(255)&CHR$(
0)&CHR$(0)&CHR$(192)&CHR$(
)&CHR$(12)&CHR$(2)&CHR$(1)&C
HR$(1)&CHR$(0)&CHR$(0)
184 PRINT #1:CHR$(254)&CHR$(
1)&CHR$(1)&CHR$(1)&CHR$(1)&C
HR$(1)&CHR$(1)&CHR$(1)&CHR$(
254)&CHR$(0)&CHR$(0)
185 PRINT #1:CHR$(255)&CHR$(
0)&CHR$(0)&CHR$(0)&CHR$(192)
&CHR$(32)&CHR$(28)&CHR$(2)&C
HR$(255)&CHR$(0)&CHR$(0)
187 NEXT I
189 PRINT #1:CHR$(13)&CHR$(1
0)
190 CLOSE #1
```

This is basically the same program except the graphic definition is 33 columns wide and spells my name. We read the codes 10 times so printing is 1-256 and 74 left over: 256 + 74 = 330, see lines 121 and 181.

All these programs print two rows for the graphics, so you can play with the line spacing in line 120 to separate the two rows.

After playing with these programs you will be able to experiment with print codes in your graphic programs to see if you can get rid of those white or dark lines on your pictures. HINT: Play with the line spacing.

Comparing pixel editors

Picasso Publisher V2 and TI-Artist Plus

By ALF RUGGERI

The author is a member of TIsHUG, the Sydney, Australia, users group. The article is reprinted from the June 1995 BUG-Bytes, *newsletter of the Brisbane Users Group.*

In this article, neither a detailed comparison between Picasso Publisher V2 and TI-Artist Plus, nor a review or user walk-through is intended. These topics have been more than adequately featured as articles in past newsletters, MICROpendium, etc.

Given the length of time since the products were first released, it is fairly conceivable that any potential users would have long since bought the same, assimilated the fairly descriptive documentation supplied, and no doubt had excellent service from them, however a particular feature requires further comment.

PIXEL EDITORS

So what purpose is this article meant to serve? Although the two products have very similar properties, the only performance overlap is in their capacity as graphic or pixel editors of 256x192-pixel images.

Pixel status adjustment as used in "touching up" scanned images can be a time-consuming and extremely daunting task, if not entirely an exercise in masochism. Therefore, if enthusiasm for creative graphic production is to be maintained, the most expedient method to reduce the mundane process must be utilized.

USING TI-ARTIST PLUS AS A PIXEL EDITOR

The procedure is:

a) Access the ARTIST option from the SELECT MENU.

b) Load an image to be processed.

c) Set the PLOT/ERASE icon for the intended task.

d) The individual pixels need to be identified, and, for this to take place, the ZOOM facility has to be set.

e) The cursor is advanced to the appropriate pixel area via the joystick or keyboard arrow/FCTN key combination. The pixel is adjusted via the fire button or ENTER key.

f) The status of the PLOT/ERASE facility can be toggled in the zoom mode by pressing "FCTN ." or simply "."

g) The ZOOM mode is canceled by a Z keystroke in order to observe the overall effect of the pixel status alteration.

h) If further pixel alteration is required, the sequence from c to g is repeated.

USING PICASSO PUBLISHER AS A PIXEL EDITOR

The procedure is:

a) From the title screen press "FCTN =" to access the FILE UTILITY MENU.

b) Load an image to be processed.

c) Set the draw/erase mode for its intended task via a U keystroke.

d) The individual pixels need to be identified, and, for this to take place, the ZOOM facility has to be set.

e) The cursor is advanced to the appropriate pixel area via the joystick. The pixel is adjusted via the fire button.

f) The status of the draw/erase facility can be toggled in the ZOOM mode via a U keystroke.

g) The ZOOM mode is canceled by a D keystroke in order to observe the overall effect of the pixel status alteration.

h) If further pixel alteration is required, the sequence from c to g is repeated.

THE COMPARISON

Both sequences have exactly the same number of keystrokes and steps and, therefore, offer little choice by way of a method shortcut between the two.

There are, however, two major advantages that Picasso Publisher has over TI-Artist Plus. They are:

1. The ZOOM mode activation and cancellation performance time in Picasso is instantaneous.

The ZOOM mode performance time for TI-Artist Plus is 10 seconds for activation and two seconds for cancellation. The manufacturers are aware of the delay, and in the documentation ask the user to be patient, but having to wait for a total 12 seconds turnaround between numerous pixel "touch-up" operations is not very appealing.

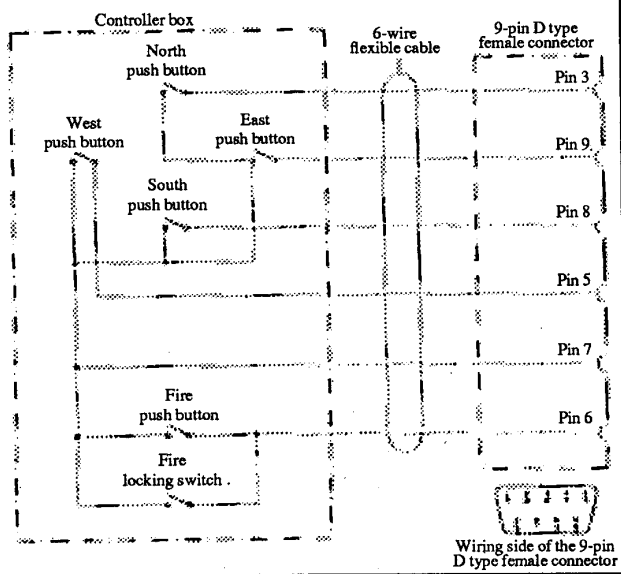
2. In the ZOOM mode of Picasso Publisher, the status and location of pixels in the magnification area are clearly identified individually in a matrix grid.

The ZOOM mode of TI-Artist Plus displays the presence or absence of pixels by areas of black or white. It is not an easy task to recognize individual turned-on pixels, not to mention those that are turned off. Not too many of us have sufficiently calibrated vision that allows recognition of discrete areas as pixel occurrence and nonoccurrence.

This ambiguity is particularly noticeable in the apparent different line widths of TI99/4A screen display areas that are

(See Page 18)

An alternate joystick, or a better controller for Picasso Publisher V2 prepared by Alf Ruggeri. Refer to the accompanying article describing the controller's derivation and purpose.



PIXEL EDITORS —

(Continued from Page 17)

assigned to linearly consecutive pixels arranged as vertical or horizontal lines.

The screen display problem mentioned in the previous paragraph is certainly a result of the 99/4A's limitations, and TI-Artist Plus copes with the problem as best as it can. Objectively, the screen distortion situations must be related to the fact that the TI99/4A was primarily designed for the NTSC system as used in the USA, not the PAL system used in Australia. I have not seen a screen display produced by an NTSC system, so I cannot make a further comment on the subject.

On the other hand, Picasso Publisher's matrix grid approach very elegantly avoids the problem altogether.

A CRITICISM OF PICASSO PUBLISHER

In spite of my (I hope) objective appraisal of Picasso Publisher as the better pixel editor (and that appraisal is based on very intensive use of both products since they were made available), not all is a source of joy with Picasso. It is sadly lacking two very useful features available in TI-Artist Plus, notwithstanding the difference in ZOOM mode performance.

The two missing features elate to cursor location management and are:

1) Single-step control of the cursor speed.

2) Keyboard control of the cursor's movement.

The absence of the first features is in part offset by the many increments of "-" and "+" inputs (65,000 are quoted in Picasso's documentation), but all the same it is very easy to lock cursor mobility, and then it becomes necessary to gingerly press the "+" key *x* number of times to restore movement. TI-Artist's approach of 10 discrete speeds set by keystrokes "1" to "0" is definitely more comfortable.

I would advise that Picasso's lowest speed be initially set up and used right throughout the PIXEL EDIT operation. This setting should be suitable for most operations, and it will certainly provide the most accurate pixel location seeker in what is really not a 256x192- but part of a larger 480x336-pixel screen.

The absence of the second feature is definitely a design oversight. Whereas all

In spite of my (I hope) objective appraisal of Picasso Publisher as the better pixel editor, not all is a source of joy with Picasso. It is sadly lacking two very useful features available in TI-Artist Plus.

but the directional arrow keys E and X are already assigned single-key Picasso function, all the arrow keys could have been utilized as in TI-Artist via their conjunctive use with the FCTN key.

The concept of the joystick facility used by Picasso was quite sound and it effectively maximized the TI's input resources, but the accuracy of the TI joysticks appliance (not the actual joystick port circuitry or software) was only intended for games playing, where the element of luck and furtive manual twitching is likely to mask the joystick's lack of positional certainty.

AN ALTERNATIVE PICASSO CONTROLLER

I was quite motivated for my GREETING CARDS presentation and subsequent article publication in TIsHUG's November 1990 newsletter, to find a better way of controlling the cursor. I tried Larry Saunders' microswitch joystick, but, in spite of better control, it still took a lot of concentration to ensure that a vertical or horizontal line when drawn would not infuriatingly veer diagonally off the intended path.

Ultimately I decided to scrap the joystick control altogether. I replaced it with four directional push-button switches, a "fire button" push-button switch, and a single-pole, double-throw toggle switch wired across the "fire button" switch to advance the cursor in a locked draw or erase mode.

I determined the joystick connections by disassembling a TI joystick and checking with an ohmmeter, the continuity of the exposed membrane switch elements back to the joystick cable's nine-pin D-type female connector.

Since 1990 I have built two of them, the latter one with a few specific features that allow extremely accurate pixel manipulation. The unit described above and in the accompanying circuit diagram is for the simpler configuration. Picasso steered via my simple controller allows as effective a key-activated control as TI-Artist Plus.

If you plan to build the controller, keep in mind that the four directional and "fire button" push-button switches are going to be subject to a lot of wear and tear. Unless you want a poor imitation of a TI joystick, do not buy the switches in blister multipacks from suburban budget outlets; quality certainly counts. The cost for the complete unit should not come to more than $30.

The plastic box in which the six switches are mounted can be as small as can comfortably house the components yet allow ease of switch operation. The length of cable can be anything up to the odd meter length of the standard TI joystick cable, although 350mm should be sufficient.

A word of warning: avoid bad soldering techniques when connecting the switches and especially the terminals on the back of the nine-pin D-type female connector, unless you enjoy coming to grief. Neither I, nor this newsletter article nor TIsHUG will be held responsible for adverse results to your system in the event of faulty construction practice.

A CONCLUDING THOUGHT

My simple controller is not presented as an inspired solution to what I consider a software oversight, but rather as a simple improvisation to enhance a very useful program. Without backtracking through endless TI documentation, I am sure that similar solutions have been considered and built, if not already published, by other users.

As a matter of fact, this entire article, on an agreeably dated topic as it is, is presented in the interest of continued support to dedicated TI99/4A users.

The contents of this article were presented by me as a mini workshop at the May 1995 Sydney TIsHUG meeting. I was encouraged by the feedback from the members in attendance to share the article with the readers of this newsletter.

Disk drives

Random access, yes, but it's not done randomly

By JIM NESS

This is reprinted from the June 1995 issue of VAST News, *newsletter of the Valley of the Sun TI Users Group in Phoenix, Arizona.*

The great thing about disk drives is that they can find files buried randomly within a huge field of data, and they do it pretty fast. Actually, they can do it so fast because it's not at all random.

The mechanical concept is not all that complicated. A small motor spins at 300 rpm (at least in this country, with its 60 hz power supply), and there is a tiny stepping motor attached to a read/write head. A stepping motor is a common item in indexing applications, where you want a motor to move a precise distance and stop on a dime. The read/write head is just a smaller version of what you have on a cassette recorder.

STEPPING MOTOR

The stepping motor "steps" the head from track to track on a diskette. The tracks are concentric circles, not a long spiral as you would have on an album.

All of this is ultimately controlled by the disk software provided with your computer. Usually this is located in ROM within the machine. In most machines, the ROM is only sophisticated enough to load in the official Disk Operating System (DOS) which is located on the disk in the drive when the machine is turned on. The DOS contains all the file handling software, copying software, etc., and, because it is on disk, it can be easily modified and/or updated as time goes by.

Our friends at TI decided to put the whole thing in ROM, which has a few bad side effects. First, it makes it hard to update ond improve the software, which is located in the disk controller card. Second, although the machine is a 64K machine, TI set aside so much memory for special purposes, that only 32K remained to play with. TI set aside 8K for cartridges, 4K for disk drive, 4K for RS232/PIO cards, 4K for the Operating System (can't complain about thot one), and 8K for various interfaces (speech, sound, VDP). Okay, those are a11 good applications to have, but if you don't use them, you still can't use that memory for other things.

Anyway, all of the controlling software for the TI99/4A is located in the ROM card, as I said. This software tells the step motor when to step to the next track, when to return to the beginning, etc.

NO STANDARD FOR KEEPING TRACK OF DATA

There is no standard for how a computer keeps track of data. In the case of TI. there is a directory of existing files, and a map of where they are located, at the beginning of each disk. These files are not necessarily all in complete groups. If you delete a 12-sector file from a disk, there is a 12-sector gap recorded in the map. Then, if you add a 20-sector file, the software will put the first 12 sectors in the gap, and put the rest in the first auailable spot. When you ask for a file that is broken up this way, you can hear the disk head srooting along to read each individual segment.

Because the disk drives themselves are pretty standard, there are a few things that don't change. For instance, there are 48 tracks per inch in most 5.25-inch systems. And most systems only use 35 or 40 of the available 48 tracks. There are either 9 or 18 sectors per track (single- or double-density). Each sector holds 256 bytes of data. And the standard design allows 250,090 bits per second to be written.

Wow, you say, 250K! That is about 25K bytes per second, right? How come I cannot load a 25K program in one second?

Two reasons. First, as I said, the transfer of data is actually controlled by the ROM software in the TI99/4A. And to be as good as it is, it had to be a little bit slow. Not real slow (anyone ever use a Commodore 64 disk drive?), but not as fast as it could be. The second reason also has to do with software, but it is a uniuersal problem associated with single-density storage.

The major difference between single- and double-density storage is the way in which the data is coded. In order for the software to keep track of where the read head is located on a particular track, there are clock or synch bits laid down with the data bits. In the old fashioned single-density format, o synch bit was laid down ahead of each "0" bit, so there were never two 0 bits in a row. That kept the software from getting lost if there were a lot of "0" bits in series. Putting all those synch bits on the disk took up a tremendous amount of space that should be used for data.

ENCODING CLOCK BITS

So, some genius came up with a way of encoding the clock bits in with the data bits, so that no unnecessary space was lost. Voila, double-density storage was born And double density, as used with the CorComp software, is said to increase transfer speed by at least 80 percent, mostly because the number of bits to transfer is cut way down.

So much for the exciting story of double-density versus single-density. How about double-sided versus single- sided? Well, obviously, it requires two read/write heads in the drive. Did you know that when reading a disk, the software reads, first, a track from side one, then the opposing track from side two, and continues back and forth?

The disk head needs something to keep the disk stationary against it. In a single-sided drive, there is a snall arm holding the back side of the disk against the head. In a double-sided drive, that arm would be in the way of the back side read/write head, so the solution was to use the two heads, directly across fron one another, to hold the first available spot. In order to keep them across from one another, they alternate reading or writing, as noted above.

Guess the secret word when you play JOTTO:5

By W. LEONARD TAFFS

The JOTTO:5 program appeared in the newsletter of the Southwest Ninety-Niners.

JOTTO:5 is a "Secret" Word Game where you try to guess someone else's five-letter "secret" word. I'll explain more later.

JOTTO:5, for TI Extended BASIC, makes it possible to play the game without having to use pencil and paper.

When run, the game takes some 25 seconds to read its word list of 148 "secret" words. Once the words are read into an array you are prompted to enter a word. Use lowercase letters when entering words. The computer scans this word and then displays on the upper screen how many letters there are in your test word that are in the computer's secret word. For example, "3table" means there are three letters in the word "table" that are used in the secret word. If any of the letters of your test word occur in the exact position of the test word, this is shown in the five underline character display, to the right of which is a display of the number of turns you have taken.

Once you know definite letters to eliminate, you can enter "xx" as a test word — this will not count as a turn — and this will bring you to a prompt to enter letters to discard. When discards have been entered, the screen displays three new lines at the top: the full alphabet at the top, followed by a line that repeats the alphabet, and then the letters you entered to be eliminated. These letters are now deleted in the second line alphabet. This helps you focus on what letters are left to choose from in your next test words.

As you enter your test words they will be displayed below the above lines in a continuing string. This allows you to deduce what letters, in conjunction with observing the 5-underline display, might be the next best choice of test word. When you have used 20 tries, a line will be displayed (line 420) asking if you want to see the word (if you're giving up!). If you would rather play the game without this interruption, you can REMark this line. You can also enter a question mark as a test word at any time in the game and the secret word will be displayed at the top of screen.

Once you have guessed the test word, the screen will turn red and you will be told the number of tries it took you to find the word.

Other options exist for your convenience in viewing the computer's word list. Using uppercase letters you can enter "LIST". This will bring up the prompt "0=SCREEN 1=PRINTER". Enter 0 (zero) to see words on screen or 1 to send the list to your printer. When you choose the "LIST" option your words will appear in the scrambled order as they are listed in the program DATA statements.

You can also alphabetize (sort) the list of words. To do this, enter (again in uppercase letters) the word "SORT," as a test word. It will take two or more minutes to do so (numbers will appear next to the word "Sorting...." so you know the program is at work).

After the list is sorted you are brought to a new menu which gives you the option of seeing them on screen or sending them to your printer or returning to the game. NOTE: Be careful after selecting option 1 or 2, as the next prompt will be asking if you wish to save the sorted list to disk as well. This feature was added so you could have a file of your sorted words. Entering "N" for "NO" will bypass the file creating option. A CALL KEY option is included (not shown on menu) which allows you at any time to escape by pressing "R."

Should you desire to leave the game to go to the sort menu, enter "yy" (lowercase) as your test word. It is possible to toggle back and forth as you wish.

A final option requires un-REMarking a portion of a line in the program. This is to un-REMark the "You have found.." portion of line 400. This will make the game easier to play for youngsters or less experienced players.

The running time of this game can be considerably shortened by REMarking some of the word list DATA statements. Twenty-five words would be plenty to use for starters. For "professional" Jotto players, I have a version with almost 300 words. You will need to adjust the LOOPs in lines 70 and 1260 to reflect your word count. Reminder: Be sure to allow for "END" as the last word.

I wish I knew who to credit the original program to. However, I have added so much to the program that it hardly resembles the original. I have added QUICKSORT to sort the words. The "scratch sheet" program (lines 1600-) are another part of my addition. This feature was not in the original game. To play the game as I found it, you still had to use paper and pencil.

I have run and played this game hundreds of times and experienced no problems with it. Incidentally, only once in plays did I ever happen to guess the secret word as my first test word! Hope you have fun with this version! Please let me know if you encounter any bugs with it.

You can always substitute or change words in the DATA statements. If you add or take away words, you must be sure to set loop/counters in lines 70 and 1260 to agree or you will get program errors preventing the game from working.

HOW TO PLAY JOTTO

For people who like word games, Jotto is an easy game to play and really comes in handy when there is time to kill — such as when making long trips in the car or waiting for a late plane arrival. All you need is scrap paper to write on and two or more people to play.

You begin by asking another player a "test" word. That person must tell you how many letters in your test word are in his secret word you are trying to guess. Words that have double letters in them make the game very difficult, so it's best to have a rule that no such words are allowed as "secret" words. However, words with double letters in them are allowed as

(See Page 21)

JOTTO:5 —

(Continued from Page 20)

"test" words.

For example: Suppose the secret word is "party." You happen to ask the test word "yeast." The opponent must tell you that the test word has three letters. In this case, you've been lucky. Had you asked the test word "lemon," the other player would have had to tell you the test word had no letters. This can also be lucky as it tells you right away five letters that can be eliminated from future test words. The test word "paper" means your opponent must tell you that paper has four letters in it: 1 for A, 1 for R, and 2 for the repeated Ps.

If you are not using the computer version of this game, it is suggested that you write out the alphabet so that you can cross off eliminated letters. Also, write out all your test words, crossing out the letters you know have been definitely eliminated. Players must be careful in not making a mistake when they tell a person how many letters in the test word are in the secret word. Slipping up on this can ruin the game.

The first person to guess the secret word wins. Often good sports will allow the other player to continue until they will find their word, too.

This can be an excellent game to improve one's spelling and vocabulary.

JOTTO:5

```
1 REM [JOTTO:5] Adaptation
 of a non-working program.
 Author unkown. 11-9-94
 Enhanced by W.L.Taffs
 SW99ers, Tucson, Az. !151
2 REM!154
3 REM UnRem last part of lin
e 400 to make second display
 show if you wish to make ga
me easier. !154
4 REM!154
5 REM Words are at 1400-1590
      610-and 1270- DELETE
      ;" "; if you desire
      single items in column
 !003
6 REM!154
7 REM LIST LOOP SET:
      line   70 I=148
      line 1260 H=148 !174
10 REM!154
20 CALL CLEAR :: DISPLAY AT(
6,3):"The 5-Letter WORD game
": :"      Enhancement by":
:" W.Leonard Taffs, SW99ers"
 !155
30 DISPLAY AT(20,5):"Reading
 Word list ""JOTTO:5""" !040
40 RANDOMIZE !149
50 REM 148 WORDS plus "END"
!037
60 DIM W$(150)!211
70 FOR I=1 TO 148 :: READ W$
(I):: DISPLAY AT(23,12):I !1
26
80 IF W$(I)="END" THEN NN=I-
1 :: GOTO 100 !088
90 NEXT I :: NN=148 !035
100 G=0 :: CALL CLEAR !081
110 S$=W$(INT(RND*NN+1))! GE
T RANDOM GUESS !085
115 DISPLAY AT(24,1):RPT$("
",28)!174
120 B$="abcdefghijklmnopqrst
uvwxyz" !036
130 DISPLAY AT(1,1):B$ ! :SP
X$ !189
140 DISPLAY AT(18,2):"Use LO
WER case letters to  guess
a 5-Letter Word.": :!148150
ACCEPT AT(21,10)SIZE(-5):L$
:: IF L$="" THEN 150 :: CT=1
 :: IF CT=1 THEN L1$=L$ :: C
T=0 !200
160 IF L$="LIST" THEN GOSUB
560 :: GOTO 140 !072
170 IF L$="SORT" THEN GOSUB
680 :: GOTO 140 !204
180 IF ASC(SEG$(L$,1,1))=63
THEN 240 !054
190 IF ASC(SEG$(L$,1,1))<97
THEN 150 !228
200 !!131
210 IF L$="xx" THEN GOSUB 16
00 !177
220 IF L$="yy" THEN CALL CLE
AR :: GOTO 1120 !035
230 ! IF L1$="yy" THEN CALL
CLEAR :: GOTO 1080 !078
240 IF L$="?" THEN DISPLAY A
T(5,2):"The WORD was: ";S$ :
: GOTO 460 !169
250 !!131
260 IF LEN(L$)<>5 THEN DISPL
```

(See Page 22)

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, Grossbauerstr. 24, A-1210, Vienna, Austria.

TI New England Fall Faire, Sept. 30, Emanuel Lutheran Church, 200 Greenwood St., Worcester, Massachusetts. Contact Jim Cox, 905 Edgebrook Dr., Boylston, MA 01505 or (508) 869-2704.

OCTOBER

Chicago International TI Faire, Oct. 28, Evanston Public Library. Contact Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009, or Hal Shanafield, (708) 864-8644.

1996 TI FAIRS

FEBRUARY

Fest West '96, Feb.17, Ramada 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.

JOTTO:5 —

(Continued from Page 21)

```
AY AT(24,1):" Should be 5 le
tters!" :: G=G-1 :: GOTO 110
 !168
270 G=G+1 ! count guesses !2
42
280 IF L$=S$ THEN 450 !101
290 M=0 :: Q=1 :: P$="_____"
 :: A$=P$ !003
300 FOR I=1 TO 5 :: J=0 !191
310 FOR J=1 TO 5 !061
320 X$=SEG$(S$,I,1)!159
330 IF X$<>SEG$(L$,J,1)THEN
370 !202
340 P$=SEG$(P$,1,J-1)&SEG$(L
$,J,1)&SEG$(P$,J+1,5-J)!078
350 IF I=J THEN A$=SEG$(A$,1
,J-1)&SEG$(L$,J,1)&SEG$(A$,J
+1,5-J)!166
360 M=M+1 !019
370 NEXT J !224
380 NEXT I !223
390 L2$=L2$&STR$(M)&L1$&" "
!072
400 DISPLAY AT(5,1):L2$ ! :
:"You have found";M;" Letter
s:":,P$ !215
410 DISPLAY AT(14,1):"So Far
 you know: .....":,A$;" #";G
 !140
420 IF A$=S$ THEN 450 !090
430 IF G=20 THEN DISPLAY AT(
16,1):"If you wish to see wo
rd, type ""?""" !211
440 GOTO 140 !219
450 CALL SCREEN(10):: DISPLA
Y AT(19,1):RPT$(" ",56):: DI
SPLAY AT(14,1):"You Got It!
 It took you";G: :" tries to
 find -> ";S$;" <-" !235
460 PRINT :: INPUT "Play ano
ther game Y/N ? ":Q$ :: CALL
 SCREEN(15)!217
470 L1$,L2$,LC5$,SPX$=" " ::
 TR,RR=0 !063
480 IF Q$="Y" THEN 100 ELSE
IF Q$="y" THEN 100 ELSE PRIN
T : :"THANKS FOR PLAYING! Le
onard." :: STOP !199
490 FOR I=1 TO 4 !059
500 FOR J=I+1 TO 5 !071
510 LET X$=SEG$(L$,I,1)!037
520 IF X$=SEG$(L$,J,1)THEN D
ISPLAY AT(20,1):"You can't r
epeat any letters": :"Bonus
Try.  Another word:" :: RETU
RN !162
530 NEXT J !224
540 NEXT I !223
550 RETURN !136
560 REM Choice to list words
 !051
570 INPUT " 0=SCREEN  1=PRIN
TER ":Z :: IF Z<0 OR Z>1 THE
N 570 !087
580 IF Z THEN OPEN #1:"PIO"
!139
590 FOR I=1 TO NN STEP 10 !1
83
600 FOR J=1 TO 10 :: IF I+J-
1>NN THEN 660 !235
602 CALL KEY(0,K,S):: IF S<>
1 THEN 610 !106
604 IF K=82 OR K=114 THEN IF
 PRN THEN CLOSE #1 :: PRN=0
:: ELSE CALL CLEAR :: GOTO 1
10 !140
608 CALL KEY(0,K,S):: IF S<>
1 THEN 608 !104
610 PRINT W$(I+J-1);"  ";!00
5
620 IF Z THEN PRINT #1:W$(I+
J-1);"  ";!096
630 NEXT J :: PRINT :: IF Z
THEN PRINT #1 !161
640 NEXT I !223
650 PRINT :"        End of 1
ist": :!125
660 FOR DLY=1 TO 300 :: NEXT
 DLY :: CALL CLEAR :: IF Z T
HEN CLOSE #1 !055
670 RETURN !136
680 REM **** QUICK SORT ****
A faster sort than most XB
sorts for a scrambled list
but quite slow for a list
nearly sorted already.  ****
 !161
690 CALL CLEAR :: DISPLAY AT
(12,5):"Sorting:" !212
700 P=1 !008
710 LL(P)=1 !013
720 R(P)=NN !105
730 IF P<=0 THEN 1110 !029
740 LB=LL(P)!161
750 RB=R(P)!097
760 P=P-1 !026
770 IF RB<=LB THEN 730 !120
780 I=LB !149
790 J=RB !156
800 T$=W$(I)!103
810 IF J<1 THEN 860 !095
820 IF T$>=W$(J)THEN 860 !13
2
830 J=J-1 !014
840 DISPLAY AT(12,15):J;" ";
I !167
850 GOTO 810 !124
860 IF J>I THEN 890 !205
870 W$(I)=T$ !103
880 GOTO 1010 !068
890 W$(I)=W$(J)!033
900 I=I+1 !011
910 IF I>NN THEN 950 !091
920 IF W$(I)>=T$ THEN 950 !2
21
930 I=I+1 !011
940 GOTO 910 !224
950 IF J<=I THEN 990 !238
960 W$(J)=W$(I)!033
970 J=J-1 !014
980 GOTO 820 !134
990 W$(J)=T$ !104
1000 I=J !081
1010 P=P+1 !025
1020 IF I-LB>=RB-I THEN 107
 !229
1030 LL(P)=I+1 !023
1040 R(P)=RB !097
1050 RB=I-1 !087
1060 GOTO 770 !084
1070 LL(P)=LB !161
1080 R(P)=I-1 !210
1090 LB=I+1 !080
1100 GOTO 770 !084
1110 PRINT :"Sorting Finishe
d." :: FOR DLY=1 TO 300 :: N
EXT DLY :: CALL CLEAR :: SRT
X=1 !036
1120 IF SRTX THEN 1130 ELSE
1140 !126
1130 DISPLAY AT(8,5):"SORTIN
G finished ";W$(1): :"
  CHOOSE:": :" 1  List on Sc
reen": :" 2  List to Printer
": :" 3  Back to Game" :: GO
TO 1150 !090
1140 DISPLAY AT(8,5):"SORTIN
G NOT DONE ";W$(1): :"
  CHOOSE:": :" 1  List on Sc
reen": :" 2  List to Printer
": :" 3  Back to Game" !244
1150 ACCEPT AT(20,13)VALIDAT
E("123"):RN !046
```

(See Page 23)

JOTTO:5 —

(Continued from Page 22)

```
1160 CALL CLEAR !209
1170 ON RN GOTO 1200,1190,11
80 !128
1180 CALL CLEAR :: GOTO 110
!016
1190 OPEN #1:"PIO" :: PRN=1
!039
1200 REM ** PRINT SORT ** !1
51
1210 CALL CLEAR :: PRINT "Do
 you wish to SAVE to Disk?":
 : :: INPUT "Y/N ":YN$ :: PR
INT :: IF YN$<>"Y" AND YN$<>
"y" THEN 1250 !015
1220 PRINT :: INPUT "ENTER F
ile Name: ":FN$ :: PRINT ::
INPUT "Enter Disk # ":DSC$ :
: PRINT :: FN$="DSK"&DSC$&".
"&FN$ :: PRINT :: SV=1 !106
1230 PRINT "Saving as: ";FN$
: :!250
1240 INPUT "O.K.? ":OK$ :: I
F OK$<>"Y" AND OK$<>"y" THEN
 YN$,FN$,DSC$="" :: SV=0 ::
 GOTO 1210 !081
1250 INPUT "Press <ENTER> to
 PROCEED":K$ :: IF SV THEN O
PEN #3:FN$,OUTPUT !237
1260 FOR H=1 TO 148 !165
1270 PRINT W$(H);" ";!028
1280 IF PRN THEN PRINT #1:W$
(H);" ";!236
1290 IF SV THEN PRINT #3:W$(
H)!087
1300 CALL KEY(0,K,S):: IF S<
>1 THEN 1330 !061
1305 IF K=82 OR K=114 THEN I
F RN=1 THEN CALL CLEAR :: GO
TO 1130 !049
1310 IF K=82 OR K=114 THEN I
F PRN THEN CLOSE #1 :: PRN=0
 :: CALL CLEAR :: GOTO 1130
!011
1320 CALL KEY(0,K,S):: IF S<
>1 THEN 1320 !051
1330 NEXT H !222
1340 PRINT :"End of List.  P
.E.T.C." :: IF SV THEN CLOSE
 #3 !064
1350 CALL KEY(0,K,S):: IF S<
1 THEN 1350 !145
1360 IF PRN THEN PRN=0 :: CL
OSE #1 :: CALL CLEAR :: GOTO
 1110 !098
1370 !!131
1380 IF RN=1 THEN CALL CLEAR
 :: GOTO 1140 !166
1390 RETURN !136
1400 REM ** WORDS ** !081
1410 DATA wrack,snafu,wrong,
yacht,young,yodel,zerox,zilc
h !252
1420 DATA whale,while,whole,
spate,waste,white,vouch,vowe
l !221
1430 DATA beaux,braid,sumac,
thyme,sepal,petal,stoma,peda
l !140
1440 DATA towel,gonad,sling,
snort,slide,slate,skate,smea
r !193
1450 DATA abide,beams,horse,
imbed,index,irate,joker,juic
y !107
1460 DATA peach,teach,hasty,
haste,march,marsh,scary,worm
y !170
1470 DATA metal,modal,ought,
paint,paste,pearl,penal,phon
e !158
1480 DATA plant,guave,group,
graze,grope,grape,gripe,gabl
e !155
1490 DATA decal,amber,aisle,
saint,aught,flunk,baste,azur
e !124
1500 DATA house,touch,mouth,
imbue,anger,avoid,squib,slop
e !205
1510 DATA cough,caste,fetch,
sable,chart,cream,crank,devi
l !084
1520 DATA adobe,yeast,input,
point,print,panic,spire,pran
k !210
1530 DATA dowel,dough,drink,
false,fetus,fetal,faint,froz
e !160
1540 DATA spout,wield,apish,
capon,coral,copra,cobra,floc
k !146
1550 DATA lunar,mauve,noise,
plaid,quoit,divot,ducat,duca
l !186
1560 DATA douse,rouse,louse,
drape,drupe,cigar,patch,matc
h !165
1570 DATA stair,chaos,moxie,
woman,xylem,alone,ruled,pros
e !216
1580 DATA stoic,azote,anole,
arose,leach,rebus,plank,prou
d !172
1590 DATA fling,flung,thong,
apron,END !161
1600 REM Remove selected let
ters "Scratchpad" line !221
1610 IF TR=1 THEN 1630 !191
1620 TPX$="abcdefghijklmnopq
rstuvwxyz" :: SP$=" " :: TR=
1 !173
1630 DISPLAY AT(1,1):TPX$ !1
23
1640 IF DK=1 THEN 1790 !072
1650 DISPLAY AT(22,10):"Ente
r DISCARD letters:" :: ACCEP
T AT(24,15):LC2$ :: Q=LEN(LC
2$):: RR=0 :: DISPLAY AT(22,
10):RPT$(" ",28):: LC3$=LC3$
&LC2$&LC$ :: DK=1 :: LC5$=LC
5$&LC2$ !192
1660 FOR Z=1 TO Q !160
1670 Z1$=SEG$(LC2$,Z,1):: LC
$=Z1$ :: GOSUB 1690 !101
1680 NEXT Z !240
1690 P2=POS(TPX$,LC$,1)+1 ::
 TPX=LEN(TPX$)!018
1700 IF P2<=1 THEN 1750 :: T
PX1$=SEG$(TPX$,1,P2-2)!103
1710 TPX1=LEN(TPX1$)!126
1720 TPX2$=SEG$(TPX$,P2,TPX+
1):: TPX2=LEN(TPX2$)!022
1730 TPX$=TPX1$&SP$&TPX2$ !1
84
1740 DISPLAY AT(2,1):TPX$ !1
24
1750 IF Z5 THEN Z5=0 !009
1760 DISPLAY AT(3,1):LC5$ !0
69
1770 RR=RR+1 :: IF RR<>Q+1 T
HEN 1780 ELSE 1790 !183
1780 RETURN !136
1790 DK=0 :: GOTO 130 :: RET
URN !163
1800 CLOSE #1 :: END !164
1810 CT=CT+1 :: READ A$ !037
1820 PRINT CT;" ";SEG$(A$,1,
5)!139
1830 GOTO 1810 !104
```

GENEVE

Use Guru Meditation to help debug assembly programs

By MICHAEL ZAPF

Suppose you want to write some machine language programs. Normally, it takes several passes until the program finally assembles — after having found the last syntax error. After the first "0000 ERRORS" you load and run it with great expectations — and nothing happens, at least not what you wanted the program to do. Now, if you are lucky, you can press QUIT (or CTRL-ALT-DEL) for a "warm" boot, and maybe the program is still in RAM so that you can PEEK it via TI-BASIC. Otherwise you have to turn your computer off and back on, losing your program in memory.

If you are using a Geneve, there is another threat not experienced by the standard TI user: System kernel RAM locations can be affected so that you cannot reset your computer any more, let alone debug your program. The reason for this is that the program counter is loaded with an unexpected address, and the execution is transferred to a part of memory which probably does not contain executable code. This happens when:

- Subprograms are not terminated correctly (RTWP instead of RT, or vice versa);
- Subprograms are not called correctly (BLWP instead of BL or v.v.);
- Array bounds are exceeded (especially with a list of addresses);
- Return addresses are lost or incorrect (forgotten DATA after BL);
- And many more.

One of the TMS9995 processor's most powerful features is the MID flag, which is not used, however, by the operating system. Whenever the processor encounters a word that is to be executed but does not represent a valid command, it generates an interrupt of level 2 which is not maskable by LIMI. The interrupt service routine can check the CRU flag 1FDA: It is set when an illegal command has been encountered. The abbreviation MID means "macro instruction detect," which reveals another usage for this feature. Currently, the MID flag is simply cleared by the interrupt service routine.

The following program is intended for the Geneve in GPL mode with standard Editor/Assembler and standard GPL interpreter. Newer versions may require changing of the absolute addresses in GROM or RAM. Assemble this program and load it via LOAD AND RUN once; it remains installed until you clear the memory completely or reload the GPL interpreter. Now take the source code of one of your programs and put a DATA >0000 right into the program text. If you run this bugged program the execution will halt right at this position and the computer will inform you that an error has occurred, giving you the values of the WP, PC and ST registers. Press the left mouse button to return to the master title screen; your program remains unchanged.

This early execution break can prevent your computer from totally disturbing the memory, but there is surely no guarantee that the computer will not hang. This program was inspired by the "Guru meditation" message of the Commodore Amiga, which is more detailed, however.

Send any questions to: zapf@rbi.informatik.uni-frankfurt.de

GURU MEDITATION

```
***     *Guru Meditation***

*       on Geneve 9640 in GPL Mode
*       Version 1.1
*       March 14, 1991
*       Michael Zapf

* Put this utility in an unused RAM space of the Geneve *
(Warning: Some extended Ed/As do use this space. Be careful! *
With the standard Ed/As, this program fits into the space * be-
tween >8100 and >8300.

AORG >8100

TEXT1 TEXT 'Software failure. Press'
TEXT ' left mouse button to continue.'
TEXT2 TEXT 'WP.PC.ST contents:'
HEXD TEXT '0123456789ABCDEF'
POINT TEXT '..'

* VDP register settings

VDPREG DATA >8004,>8170,>8203,>8347
DATA >8401,>8506,>8600,>8761
DATA >8888,>8900,>8A00,>8B00
DATA >8C06,>8D66,>8E00,>8F00

* Check MID flag. If not active, return.

START LI R12,>1FDA
TB 0
JEQ START1
B @>035E

START1 SBZ 0      * Clear flag
LWPI >8300

* Set VDP registers to defined values

LI R1,VDPREG
LOOP1 MOV *R1+,R0
BL @SETADR
CI R1,START
JL LOOP1

* Clear the screen

LI R5,>8C00
LI R0,>4000
BL @SETADR
LI R2,1920
LI R1,>2000
LOOP MOVB R1,*R5
```

(See Page 25)

GURU MEDITATION —

(Continued from Page 24)

```
D(
JNE LOOP

* Load upper and lower case character set * from GROM to VDP
(R5 is VDPWD)
* (to repair probably destroyed character definitions) * Note:
The characters are stored by seven bytes in GROM * so that a 00
must be inserted before each definition.

LI R0,>06B4       * GROM position
MOVB R0,@>9C02
SWPB R0
MOVB R0,@>9C02
LI R0,>4900       * VDP position
BL @SETADR
LI R3,96 * No. of characters
CLR R1
JL       MOVB R1,*R5       * Insert 00
LI R2,7
IL       MOVB @>9800,*R5   * Copy seven bytes
DEC R2
JNE IL
DEC R3
JNE JL

* Write texts on screen

BL @WRITE
DATA >40AC,TEXT1,56
BL @WRITE
DATA >4158,TEXT2,18

* Write values of R13,R14,R15 on screen. * R3 will be used as a
pointer to the RAM location * of these registers (starting at
>83DA). * R5 is still >8C00 (VDPWD).
* Use the character list HEXD to transfer the nybbles * (half
bytes) to the corresponding ASCII char.

DECT @>83DC
LI R3,>83DA
LI R6,3

DISP MOV *R3+,R4
LI R2,4  * four nybbles per word

LD       SRC R4,12         * shift to rightmost nybble
MOV R4,R1
ANDI R1,>000F     * mask it
MOVB @HEXD(R1),*R5 * Write char to VDP
DEC R2
JNE LD

DEC R6   * next register
JEQ FRAME
MOVB @POINT,*R5 * Print a "."
JMP DISP

FRAME BL @CLRBLK * Clear blink attributes
LI R2,6  * Six rows
LI R0,>5009 * Step forward to columns 72+

* Set blink attribute of columns 0 and 79 * of six consecutive
rows

LI R1,>0180
F1       BL @SETADR
MOVB R1,*R5       * column 79
     R1
   R1,*R5       * column 0
SWPB R1
AI R0,10 * next row
DEC R2
JNE F1

* Set blink attribute of every column in the first row

LI R2,10
LI R0,>5000
SETO R1
BL @SETADR
F2       MOVB R1,*R5
DEC R2
JNE F2

* Set blink attribute of every column in row 6

LI R2,10
LI R0,>503C
BL @SETADR
F3       MOVB R1,*R5
DEC R2
JNE F3

* Check the left mouse button. If depressed, stop blinking *
and return to the master title screen.

MLOOP LI R12,>0038
TB 0
JEQ MLOOP
BL @CLRBLK
LI R0,>8D0F
BL @SETADR
BLWP @>0000

* Standard routine to set a VDP address. * Note: Add >4000 for
writing to the address

SETADR SWPB R0
MOVB R0,@>8C02
SWPB R0
MOVB R0,@>8C02
RT

* Write the text on screen defined by the following three words:
* DATA screen pos, RAM pos, length

WRITE MOV *R11+,R0
MOV *R11+,R1
MOV *R11+,R2
MOV R11,R10
BL @SETADR
WL       MOVB *R1+,*R5
DEC R2
JNE WL
B *R10

* Initialize the blink attribute table (VDP >1000) * by writing
00 in all positions

CLRBLK MOV R11,R10
LI R0,>5000
BL @SETADR
LI R2,270
CLR R1
MOVB R1,*R5
DEC R2
JNE $-4
B *R10

* Paste the B @START into the interrupt routine * of the con-
sole ROM (which is RAM on the Geneve) * replacing the clearing
of the MID flag. * Normal execution will resume at >035E (see
above).

AORG >035A
B @START

END
```

MICRO-REVIEWS

TI99/4A Software Data Base, XB Packer V1.2, Extended BASIC V2.3

By CHARLES GOOD

TI 99/4A SOFTWARE DATA BASE by Mickey Cendrowski and Notung Software

This is a TI-Base template used to keep track of all your TI software and reference books. First you load TI-Base (available from most generic TI dealers) and then you load Software Data Base. The following are entered for each item in your inventory:

PROGRAM; the title of the software or book.

SOURCE; where you obtained the product, maybe with a date and cost.

MEDIA; a unique number assigned to the item, such as M0023. Use D for disk, M for module, C for cassette, and B for book.

CODES; a group of letters and numbers. Suggested codes are C-copyrighted, F-fairware, P-public domain, T-official TI program, S-single file program, M-multiple file program, and a number showing the total sectors the program occupies.

MODULE; the module needed to run the program, or "BAS" for TI BASIC.

TYPE; the kind of software. Fifty category abbreviations are suggested. Some examples of these categories are TOD (Tunnels of Doom games), MIDI, MAC (MacPaint pictures), TIA (Artist pictures), GRAP (graphics programs), HARD (hardware projects), etc. Help screens listing all the suggested category abbreviations and their definitions are available on line. The user can create additional software categories as desired.

COMMENTS; up to 39 characters of custom information about the program.

Appending new database items or deleting old items is easy. Searching for a program name is also easy, although you have to spell the program name exactly.

Perhaps the most useful way of displaying SDB data is in printed reports. I find that it is quicker and easier to find a program listing on a group of printed SDB pages than it is to load TI BASE, then load SDB, then have the computer scan the database for a particular program. Printouts can be alphabetical for the whole database at once, or each category separately alphabetical within the category. Printing is in condensed print (136 columns) with all the data for a particular program on a single line. The printouts are designed with a wide left margin so you can hole punch the sheets and put them in a ring binder without losing the ability to see all the data. I particularly like this wide left margin feature.

Other public domain or shareware TI disk software databases let you put each of your disks in the drive and automatically organize the data for you in ways you might not find useful. With these "automatic" databases the disk directory of each disk is sucked into the database. This lets you find all your disks that have the program LOAD or UTIL1 on them, which is often meaningless. The automatic databases also let you bring up a directory of your "disk 238," which is something you can often also do by pulling 238 out of its disk box and reading its label. A disadvantage of SDB is that it is not automatic. You have to manually type in all the data. An advantage of SDB is that you can customize the database to your specific needs, even creating your own software categories. Also SDB is the only product of its kind that lets you keep track of *all* your TI stuff, including books, cassettes, command modules, and disks.

SDB costs $20 and is sold by Notung Software. It comes on a SSSD disk with nicely written 30 page user guide. You need TI BASE v2.0 or later.

XB PACKER v1.2 by Brad Snyder

This utility allows users to make practical use of an AMS (Asgard Memory System) 128K or 256K memory expansion card. With XB Packer you can load up to 10 Extended BASIC programs into a 256K AMS card (5 into a 128K card). These XB programs all reside in bank-switched CPU memory and can be run in any of several ways. You can manually switch memory banks from XB command mode and enter RUN, or you can run the programs from XB Packer's menu by entering the number next to the program's menu listing, or let the programs automatically call and run each other.

I find XB Packer really user friendly. When you turn on your system you run the Editor/Assembler5 program ABOOT to set up the AMS card. Then you go back to the title screen, insert the XB module, and run XB Packer from XB. Once XB Packer is loaded you can use SHIFT/FCTN to change memory banks and a little display at the top of the screen tells you which of the 10 (or 5) banks you are in. From XB command mode go to bank 1 and enter OLD DSK1.XBPROG#1. Then SHIFT/FCTN to bank 2 and OLD DSK1.XBPROG#2, etc., loading XB programs into as many of the 10 banks as you want. At any time you can press SHIFT/CTRL to bring up a menu from which you can RUN a program in any bank. You can put custom titles in this menu to specify the name of the program in each bank. When you have loaded all the banks you want you can, with one command mode CALL LINK, save all the banks at once back to disk along with their menu titles in a special format. Next time you use XB Packer you can, with another CALL LINK, automatically load all the XB programs and their menu entries from disk (or RAMdisk or hard drive) back into their respective memory banks, all in one continuous operation. Thus, once you get a group of XB programs set up the way you want them it is never again necessary to load them manually one at a time into each memory bank. The autoload of a group of programs can be set up so that a program in a specified bank immediately starts running after all banks are loaded. You also have the option of displaying the menu of program titles after an autoload

(See Page 27)

MICRO-REVIEWS —

(Continued from Page 26)

so the user can select which program to run first. At any time from a running program pressing SHIFT/CTRL will get you back to the menu.

Any program in any bank acts as if it were a self-contained XB program. You can stop a running program with FCTN/4, edit the program and save it to disk, OLD in another program, etc. You can then switch banks from command mode by pressing SHIFT/FCTN and RUN or edit programs in other banks. XB Packer seems compatible with the various GRAM-based extended Extended BASICs. I have tried XB Packer with TI XB enhanced with Art Green's GUMS, with RXB v1002 and with Tony Kneer's XB v2.3, all with no obvious incompatibilities.

Probably the most practical use for XB Packer is with a series of XB programs that load and run each other, programs that contain statements like RUN "DSK1.NEXTPROG" as part of their code. To make this work, just replace each RUN "DSK1.NEXTPROG" with CALL LINK("BANK",x) where x is a memory bank number. When a running under XB Packer reaches this code, XB Packer will switch to the specified bank and immediately RUN the program in that bank. Having XB programs call other XB programs this way is fast, faster than RUN from RAMdisk and much faster than from floppy disk.

XB Packer has a couple of limitations. One — you can load *only* runnable XB programs into the AMS memory banks when using XB Packer. If your XB program reads data files or assembly language files, these files still have to be on a disk or RAMdisk. Two — XB programs run under XB Packer can't have any imbedded assembly code. If you RUN an XB program with imbedded assembly code under XB Packer the program will appear to run properly, but the assembly code will overwrite XB Packer so that bank switching is no longer possible. If a CALL LINK to another bank is attempted you will get an XB or message. This is a significant limitation. Most of the better XB packages that load parts of themselves into memory as needed have assembly calls. This is true of Bill Gaskill's Mail List Manager, which would be a great candidate for running under XB Packer except for a little bit of assembly code found in only one of its separate XB program pieces. If an XB program listing says SYSTEX near the beginning this means the program contains assembly code and probably won't properly bank switch to other programs in other banks. I hope Brad Snyder will write a version of XB Packer that puts its assembly code in low memory. This would result in less memory available for XB code but would not interfere with the imbedded assembly code contained in many of the better XB programs available today.

To use XB Packer you must have an AMS 128K or 256K card in the Peripheral Expansion Box. The AMS card and XB Packer have been successfully tested by me on a 40- and 80-column (AVPC 80-column card) 99/4A system. AMS doesn't work with a Geneve. XB Packer is part of the software package given to purchasers of the AMS card. Included is on disk documentation and a demo set of music XB programs all set up to batch load into the AMS and play from the XB Packer menu. XB Packer is fairware and the author requests only a $5 donation.

EXTENDED BASIC VERSION 2.3
by Tony Knerr

This is another "extended Extended BASIC" for GRAM devices. You need a GRAM Kracker, P-GRAM, Gramulator, or Geneve to run this. XB v2.3 adds additional features to regular TI XB and is fully compatible with the original. Any software written for TI XB will run under XB v2.3 with no problems. XB v2.3 is public domain.

I recently reviewed another similar GRAM-based public domain extended Extended BASIC called RXB. RXB is so extensive and offers so many enhanced features that one might wonder why anyone would bother considering another similar product. Well, with extended Extended BASICs (there are several of them in GRAM, module and disk additives to regular TI XB format) it is the little things that count. Which extended XB has the particular combination of additional features desired by the user? They are all a little different. If you have a hard drive or a big RAMdisk you can have several of these XBs available and switch between them as desired. XB v2.3 has some unique and useful features.

For me, the most useful feature of XB v2.3 is its compatibility with enhanced PC keyboards. When running Extended BASIC software the BREAK key (Fctn/4) now works! One of the most annoying aspects of using either a Geneve or the Western Horizons Technology XT keyboard interface is that you can't press the F4 or Fctn/4 keys to stop a running Extended BASIC program, which is something we are all used to doing on our 99/4A keyboards. Until now, when running XB software on a Geneve or when using the AT keyboard interface, the only way to abort a running XB program was to reset the computer. This is often inconvenient and causes you to lose potentially important information. XB v2.3 comes in a 99/4A version and a Geneve version. Using the Geneve version on a Geneve you just press the F4 key when running XB software and the computer screen says BREAKPOINT IN LINE xxx just like pressing Fctn/4 on a 99/4A. I haven't tried this on thWHT AT keyboard interface, but I suspect F4 should BREAK there too when using XB v2.3. I love this! XB v2.3 is now the version of XB I use most often on my Geneve.

When you start XB v2.3 you can bypass the autosearch for DSK1.LOAD with the space bar. XB v2.3 has a nice resident true lowercase character set. By using one-word CALL commands you can reset all redefined characters at once to include these lowercase characters, or to TI's original character definitions, with or without affecting screen foreground and background colors. You can, with one CALL, set all foreground, background and screen colors to any of 13 predefined and quite readable combinations. You can catalog any drive, including RAMdisks and hard drives, with a path name up to 11 characters, a very handy feature. Some of the other enhance CALLs, all of which work from within a program as well as from

(See Page 28)

MICRO-REVIEWS —

(Continued from Page 27)

command mode include: BEEP, CHIME, CRASH (makes a sound, doesn't destroy the computer), GPEEK (reads GROM addresses), HELP (lists the new commands), HONK, LRGCPS (loads the title screen's large capital characters), MLOAD (loads and runs EA5 software), MSAVE (saves part of memory as EA5), NEW (works from within a program), NYANYA (makes the sound kids make to taunt each other), SCREENOF & SCREENON (blanks screen & turns it back on), SPOF & SPON (stops and restarts all sprite motion), VPEEK, VPOKE, WAIT (causes a delay of defined length), XB (restarts XB), and the very significant XXB.

All of the above enhanced features and extra calls are available all the time when using XB v2.3 and are fully compatible with all software designed for use with regular TI XB, even software that has assembly CALL LOADs and LINKs. If you enter CALL XXB this will load Barry Traver's XXB v1.5 into lower memory giving you access to all the XB v2.3 and all the XXB v1.5 commands. XXB v1.5 has previously been available on disk CALL LOADable from the TI XB module. You don't need the Traver XXB disk when using XB v2.3 because all the XXB code is already contained in the XB v2.3 gram files you have already loaded into your GRAM device or Geneve. When you enter CALL XXB to put the XXB routines into low memory this may result in incompatibilities with Extended BASIC software that has assembly calls which also use low memory.

The additional features of Barry Traver's XXB include some of the following. You can read from and write to single sectors on a disk. Disk files can be protected or unprotected. A complete suite of 40-column text mode commands is available. You can move back and forth between graphics (32-column) and 40-column text mode and you can define the left and right margins for text in each mode. You can PEEK and POKE CPU and VDP memory using strings, which is more efficient and probably easier than numbers. This is a partial list of XXB's features.

XB v2.3 contains no special editing capabilities similar to the editing features of RXB. You can't, for example, move the cursor up/down within a line of code and you can't move or delete whole blocks of line numbers. When I want to do these things I use RXB in my P-GRAM or my Geneve. XB v2.3, like RXB, is under some circumstances not recognized by Funnelweb's Disk Review when trying to run XB software from the Disk Review disk directory.

As I said, its the little things that count, and no extended Extended BASIC does it all. I usually use regular TI XB (modified by Art Green's GUMS) in conjunction with Funnelweb's Disk Review on my 99/4A system. I usually use XB v2.3 my Geneve because the BREAK key works. I use RXB for programming and editing XB code and for running DV80 USER batch files.

XB v2.3 comes on a DSSD disk with on-disk documentation. Since XB v2.3 is public domain, all owners of GRAM devices and, in particular, all Geneve owners should have it in their library. I'll send it to you for $1. Even though it is public domain, you might want to send the author a nice letter of appreciation, and maybe also some money.

ACCESS:

Notung Software (Send $20 + $2 shipping for Software Data Base); 7647 McGroarty St., Tujunga CA 91042

Southwest 99ers (They sell the AMS card which includes XB Packer. A 256K card costs $100); P.O. Box 17835, Tucson AZ 85730

Brad Snyder (XB Packer author); 4260 Cedar Dr., Walnutport PA 18088

Tony Knerr (author of XB v2.3); 17 Marshall Circle. Downington PA 19335 Phone 610-269-7447. Compuserve #72070,513

Charles Good (send me $1 for XB v2.3); P.O. Box 647, Venedocia OH 45894. Phone 419-667-3131. Email cgood@osulima1.lima.ohio-state.edu (preferred), or good.6@osu.edu

READER TO READER

The B.C. 99er User Group is looking for 80-column cards, either Digit or TIM, working or not.

Write B.C. 99er User Group, c/o Ron Warfield, 216 10th Ave., New Westminster, British Columbia, Canada V3L 2B2.

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

NEWSBYTES

Address change

New mailing address for the Chicago TI User's Group is P.O. Box 7009, Evanston, IL 60204-7009.

UKgets TI BBS

The TI User Group U.K. has recently installed a BBS.

The board usese S&T BBS Software written by Tim Tesch and runs at 9600 baud. Phone number is 01623 491282. Sysop is Trevor Stevens, chairman of the group.

Initially, the board is running 6 p.m.-10 p.m. Fridays and 10 a.m.-10 p.m. Saturdays and Sundays (United Kingdom time). According to TI*MES, the group's newsletter, the board will not be available during August.

Navy recruits on-line

The United States Navy Recruiting Command has unveiled a new World Wide Web home page on the Internet.

The site can be accessed @ http://www.navyjobs.com., according t Lt. La'Tonya Harris-Mora, director of public service advertising for the Recruiting Command.

USER NOTES

Quick fix aids lockups

This is a tip from Hal Shanafield. It is reprinted from the *Chicago TImes*, newsletter of the Chicago Users Group:

Some time ago I saw an article about a fix for a problem I used to have: the tendency for the "elephant foot" connector to wiggle and break the connection between the P-box and the console, usually resulting in the dreaded lockup. The article advocated drilling a hole in the corner of the "foot" and using a screw to hold it to the console or Speech Synthesizer. With my lack of small-motor skills I would probably drill a hole right through the middle of some chips, all of which are needed, I presume.

For some time, as I said, I had the lockup problem. Sharing my computer with three kids and my wife, who was constantly writing papers for her various graduate courses, meant that it was always in use, and by hands of different skill levels. The console was getting pushed around a lot, despite my best efforts to keep it in one place, and the constant pressure on the "elephant foot" caused a steady stream of lockups, usually at the worst times.

For a while, my daughter was saving her files after every few words! She really didn't trust Dad's computer. Finally, I couldn't take it any more, and I hit upon a simple fix. I took six big rubber bands (which the mailman uses around a stack of my mail), and after dismounting the Speech Synthesizer from the console, stretched them around the Synthesizer and the foot so that there was a good tight fit. That left me with the problem of the wandering console.

I remembered that when my father lost his arm he had a roll of a rubberized plastic that held his plate in place while he learned to cut his food with one hand. A piece of that stuff under the console had the effect of keeping it anchored as if it were nailed in place. There it has been for more than two years, and lockups are a thing of the past.

Besides the cost, which is always important to TIers, this arrangement has the added advantage of being instantly reversible. Every now and then a rubber band calls it quits, but I have found that even one rubber band seems to be enough to keep the partners "married."

Oh, yes, the legal disclaimer: As with all technical hardware modifications, this august publication will assume no responsibility for anything you screw up. And it may void your warranty. (Remember those?) Those needing specifications and detailed schematics can send me a certified check for $100, for which price I will also include an operator's manual and an owner's registration certificate, suitable for framing.

XBASIC RND not just for games

The following was written by Walter Allum and appeared in TI*MES, the newsletter of the TI User Group of the United Kingdom.

Rather to my surprise, some experiments of mine seem to have identified the congruence underlying Extended BASIC RND:

X(N+1)=14389820420821*X(N)+21132486540519(MOD 10^14)

where X() is an integer.

The random fractions delivered are X()/10^14. The coefficients all conform to the standards set out in various authorities (e.g. Knuth "Art of Computer Programming," Vol2; Addison-Wesley 1981).

Strictly, we should submit the routine to, say, spectral analysis to verify its statistical performance. But the signs are good. The expected cycle length is 10^14, enough for over 200,000 years of continuous use!

Maybe some readers will ask, "Why this fuss about RND?" I would reply that, if your only interest was to set initial positions and govern the motions of "invaders," then forget it. There are many more serious uses. Knuth names the following: simulation of real-world processes; sampling of situations too varied to examine exhaustively; solving complicated numerical problems; testing computer algorithms; and decision-making. With these in mind, does it makes sense to crow about the superiority of our TIs over mere IBMs while uncritically accepting a facility whose specifications nobody seems to know?

IMAGE and USING

The following was written by Jim Swedlow and originally appeared in *ROM*, the newsletter of the User Group of Orange County (California).

Extended BASIC left-justifies all printed and displayed strings and numbers. While this is correct for strings, numbers should be lined up by the decimal point. One hard way to fix this is to turn your numbers into strings and add leading spaces. PRINT USING is much easier.

Enter and run this program:

```
100 FOR I=9 TO 10 STEP .1
110 PRINT USING "## ##.# #.# 
#":I,I,I :: NEXT I
```

Note how the first column prints the number rounded to the nearest whole number while the other two display decimals. The string of asterisks shows you that the number was too big for the space allotted.

Only the number on the screen is rounded — the number in RAM is not rounded.

The statement after USING can be a string, a string variable or a line number that refers to an IMAGE statement. Here are some examples:

```
10 A$="The answer is ###.##"
:: B$="John" :: C$="Dear"
20 IMAGE ## + ## = ###
30 PRINT USING A$:2.45678
40 DISPLAY AT(1,1):USING 20:
14,19,14+19
50 PRINT #1, USING "$###.##"
:23.1
60 PRINT USING C$&"#####":B$
```

For more details, look in the Extended BASIC manual. You can make your output look more professional with these tools.

Clean internal contacts, too

We reprinted this tip from *Wordplay*, the newsletter of the Portland Users of Ninety-Nines (PUNN) in Portland, Oregon. They credit Richard K. Stevens:

(See Page 30)

USER NOTES

(Continued from Page 29)

How about the old computer lockup hassle? Dirty contacts are the culprit on this one. Just about everybody cleans the external contacts, but they may still have problems. The culprit lies inside the console with the cartridge L-connector. What one needs to do is open up the console case, undo the screws (2) for the power supply, undo the screws (3) that hold the motherboard in place, disconnect the power supply and the keyboard connections and remove the cartridge connector.

Looking at the male part of the connector you will notice some indentations and black corrosion on the soldered area of the contacts. Take a piece of nylon scrub pad and buff those contacts on both sides until the indentations are gone and the contacts are smooth. Also do this to the board edge connectors for the I/O port. It is a good idea to spray the female part of the cartridge connector with contact cleaner while it is out. Now reassemble everything in reverse order (be patient and care You will now find that hardware lockups and erratic behavior will be a thing of the past.

BUGS AND BYTES

Delphi access to continue?

On-line service Delphi is reportedly testing new PC-compatible software. Its Command Line Interface (CLI), which allows modem users with TIs and Geneves to go on-line with Delphi, will stay until December 1995, but reportedly no final decision has been made as to whether CLI service will continue past that time.

Right now, Delphi has SIGs for a number of "non-standard" computers, including TI-NET for users of the TI99/4A and Geneve.

The making of a flippy

This article was written by Ed Mandich of the West Penn 99ers. It appeared in the group's newsletter.

What's a flippy? It came into the TI community as a floppy — for use in single-sided disk drives.

True, many of us have double-sided, double-density drives and have been on the highway since TI's inception. As for me, it's been about a year.

I started with the cassette recorder and recently went to disk drive — single-sided at that. I can imagine that there are still a lot of single-sided drives in use today. I

(See Page 31)

USER NOTES

(Continued from Page 30)

I've read about flippy kits that are marketed to change disks to double-sided. I changed my disks over with a single-edge razor blade and a hole-punch. The cost was 86 cents plus tax, less than a dollar.

The hole-punch was exactly what I needed, producing a one-quarter inch hole. You can find this at any office supply store.

I used a carpenter's T-square, measured down from the top of the disk to the top of the write-protect notch and measured 1 3/16 inches from the top of the disk to the top of the notch. I then measured from the top of the disk to the bottom of the notch and measured 1 7/16 inches. The notch length is exactly one-quarter inch by one-eighth inch deep.

Using the T-square, I marked the area on the back side of the disk with a fine point felt-tip pen, and cut a notch into the disk's protective jacket using the razor blade.

Again, measuring from the top of the disk, opposite head-access window, which is 1 3/8 inch by one-half inch, you get a measurement of 2 7/8 inches from the top of the disk to the center of the index hole window and 1 5/8 inches from the edge of the disk to the center of the index hole window.

Again, mark dimensions with a felt-tip pen. Any other type of pen is a a definite no-no.

Next, slip a piece of paper through the centering hole (large hole in the center of the disk) to protect the disk. Then, gently lift the disk's protective cover with your fingernail and slip the hole-punch gently into the centering hole and punch your index hole. Do the same on the other side. This index hole doesn't have to be precise, as the hole that lets the light through is one-sixteenth inch and the hole that you punch is one-quarter inch. Voila, a flippy is born!

MDOS 2.50 and autoload

The following item was written by Mike Doane and appeared in the newsletter of the Southwest Ninety-Niners of Tucson, Ariz. It has been edited to fit the available space.

I brought my SCSI card and Horizon card home from Fest '95, installed both, loaded up MDOS 2.50S and was informed that, although my SCSI drive was there, I had 0 bytes used and 0 bytes available. AARRRRGGGGHHHH!

I knew my SCSI and drive worked because I had formatted the drive. I then booted from the floppy, and the SCSI was back! I reformatted the HRD again, and it had the same problem. I pulled the batteries to erase memory and tried again. Same result.I mumbled the secrete incantations to the computer gods and when neither of those worked, I called Jack (Mathis).

What we found out was that although the FORM3MEG program had formatted the HRD and loaded the SYSTEM/SYS file, it does not update the FDR sector as the the length of the new MDOS. The old MDOS was about 488 sectors long while MDOS 2.50S is 520 sectors long. MDOS was loading correctly, but not all of it.

The fix for this is easy. When you sector edit the FDR, usually found in sector 0008, change byte No. 28 to read: 0 0 0 2 2 0.

You can change your ASSIGN command and assign your SCSI as SCS1 or 2, or whatever.

The above solution works only on MDOS 2.50S. Earlier versions don't have this problem.

CLASSIFIEDS

WANTED

WANTED

Need docs for Menu BH. Will pay up to $10. Larry Golab, 2112 Washington, Algonac, MI 48001-1049. v12n7

Support Our Advertisers!

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 ❑ ❑ ❑ (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