THE TORPET

$2.00

The INDEPENDENT Commodore Users' Magazine No. 27 MAY 1984

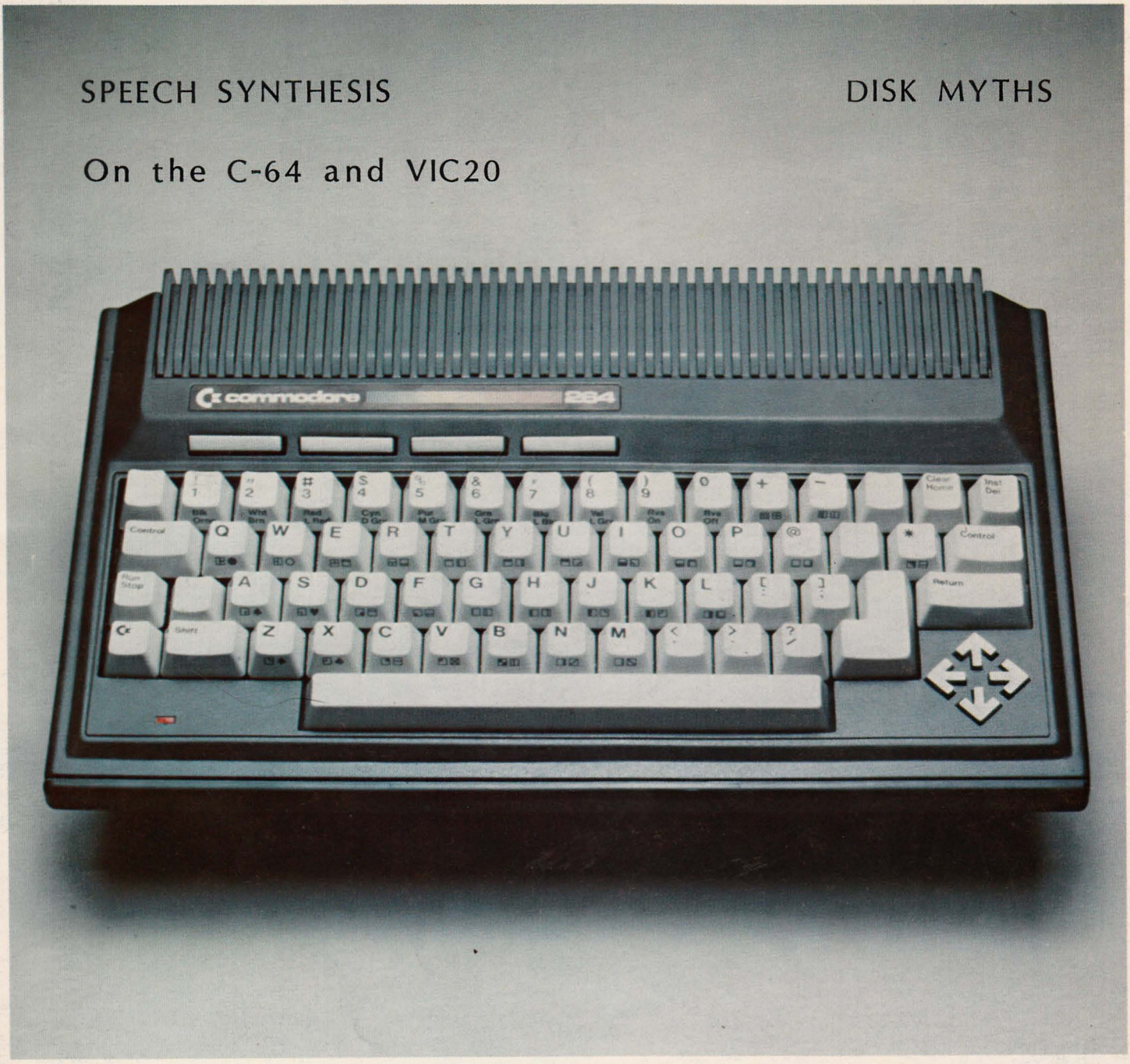
NEW DIRECTIONS FOR COMMODORE

Look Out Big Blue?

SPEECH SYNTHESIS

On the C-64 and VIC20

DISK MYTHS



Jim Butterfield Takes A Look At The New 264

THE BEST OF THE TORPET

Available From:

Copp Clark Pitman Ltd.
495 Wellington Street West
Toronto, Ontario M5V 1E9
(416) 593-9911

MORE

FOR THE COMMODORE 64* AND THE VIC-20*

1984 EDITION

EDITED BY
BRUCE BEACH

320 INFORMATIVE PAGES

*Commodore 64 and Vic-20 are Registered Trademarks of Commodore Business Machines, Inc.

TABLE OF CONTENTS

The TORPET
ISSN # 0821-1809
published by
The Publisher
Horning's Mills, Ontario
Canada L0N 1J0
(519) 925-5376
Use above address for both manuscript submission and advertising

U.S. Address: The TORPET
1 Brinkman Ave., Buffalo, N.Y., 14211

Bruce M. Beach, Publisher and Editor

Published monthly (except July and August)
Single subscription rate - $20.00 per year

Printed In Canada
Canadian Second Class Mailing Permit Number 5918
Mailed at Shelburne, Ontario
U.S. Second-Class Postage paid at Buffalo, N.Y.

POSTMASTER: Send changes of address to The TORPET, 1 Brinkman Ave., Buffalo, N.Y., 14211

LETTERS TO THE EDITOR

TOTL TEXT O.K. BUT NOT FOR COMMODORE PRINTERS

Dear Sir,

As per the review of TOTL Text (TOTL Software) in the Jan./84 issue of TORPET, I feel moved to offer a minority opinion.

Unlike Mr. Westerman, I have used decent wordprocessing programs, as well as TOTL Text 2.0 and, when (if) TOTL Text works, it is at best average. I say "if" because, unmentioned in the advertising, there is a potentially fatal bug.

TOTL Text seems not to work on Commodore printers, notably the 1515 and the 1525. The warning about this is tucked away on page 17 of the 2.0 manual. Unfortunately, what could have been a great feature, copy-ability for backup purposes, now becomes a critical liability. The dealer cannot (and will not) accept returns, even if the program will not function. By the time you read the warning, you've already bought the program, and your money is gone....

A letter from the company suggests that, for the 1525 printer, line 1150 be added as "1150 SYS64490", and contains a solicitation for US$2.00 for unspecified "software support".

It is the company's contention that the flaw lies in the timing of the Commodore printers, and claims that the software is not at fault, so "we can offer no refund or exchange". In other words, "Tough luck, sucker; next time buy a better printer."

To someone who had never used another wordprocessing program, these excuses might have been acceptable, but other programs are available, and they seem to work just fine on Commodore printers. As pappy used to say, "It's a poor workman that blames his tools."

While it is true that the VIC 20, with only a 23-column screen, will never be a truly great wordprocessing machine, a decent program can make it completely acceptable for most users. This letter is being written on "Wordcraft-20", a ROM-based program by P.L. Dowson, from United Microware Industries (UMI about U.S.$65), and it is absolutely the best WP program for the VIC that I have had the pleasure of using. As you can see, it will drive the 1525 printer adequately. Time presses, but I will try to get in a review of WC-20 to TORPET soon.

Again, if you use a Commodore printer, avoid the products of TOTL Software Inc., Walnut Creek, CA 94596.

Thanks for a forum to get this off my chest.

R.F. Tolli

16K EXPANDER FOR PET CORRECTION

Thank you for printing my submission entitled "16k Expander for PET" in the January 1983 issue of THE TORPET. I regret that I've found an error in the submission. The article implies that the expansion circuit is for an 8k PET. That is incorrect, as the pin assignments on the J4 edge connector are for adding 16k of memory to a PET with 16k of internal RAM memory.

To operate properly on a PET with 8k of internal RAM requires the following two simple changes to the pin assignment of the edge connector, J4, shown on page 54:

J4 pin 21 should be changed to read J4 pin 17
J4 pin 22 should be changed to read J4 pin 18

This is required as pin 17 is the BLOCK SEL 2 line and pin 18 is the BLOCK SEL 3 line.

I apologize for any inconvenience that this may have caused.

Sheldon H. Dean

LETTERS TO THE EDITOR

PET IN THE TORPET

Dear Sirs,
I rather object to the (new?) sub-title on the TORPET (Jan. '84): "For the Commodore 64 and VIC"

I realize that ownership is going heavily to these machines and that's okay. But those of us with PETs and SuperPETs like to think that the TORPET is for us too. The clear implication of the sub-title is that it isn't any longer.

Roger Green

REPLY

It has been the policy of The TORPET to publish every letter to the editor. However, the preceding letter is representative of others that we have received and so in this case we are making an exception.

In part the following editorial answers the question, "Will The TORPET continue to carry articles about the PET and SuperPET?" Quite honestly, it is hard to tell the future. Three months ago I would have indicated that it is less likely than I would be inclined to respond now. Things keep changing. Three months ago we were all agog about the forth coming 264. And then it was disannounced. Sort of, anyway. Who knows?

Then as the next article explains, Commodore's independent dealers started fading away and we found ourselves going in new directions of requested circulation to clubs and schools that have lots of PETs. We are on the whiplash of technological change and can never be quite sure which direction we will have to jump next.

Anyway, the bottom line is: For the time being, our interest in the PET has been revived because the percentage of our readers interested in the PET has increased due to the change in our circulation. Who knows where tomorrow will lead?

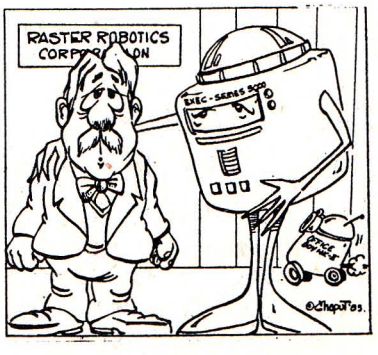
COPYRIGHT

It remains The TORPET copyright policy that any and all non-profit clubs are welcome to reprint articles from The TORPET in their magazines. If it is your policy to remmunerate authors please make out the check to the author and send it to them in care of The TORPET so that we may forward the check to them.

We also liberally grant republishing permission to other publishing organizations. For prior permission please write or call The TORPET.

WRITERS AND CARTOONISTS WANTED

The TORPET, as always, needs contributors. As a result of the split with TPUG some of the old familiar names of writers have gone in that direction leaving a vacancy in our pages that we hope will be filled by our new readers. We especially have room for new cartoonists and regular cartoon strips. We pay a minimum of $20 per page for stories and proportionally for art. Photographs, illustrations and drawings are especially welcome. Please make your submissions to: The TORPET, Horning's Mills, Ontario, L0N 1J0 Canada.



Sorry to have to let you go.
You have been with us a long time.

NEW DIRECTIONS FOR THE TORPET

A later article is entitled "New Directions For Commodore", and in this changing computer world The TORPET is also off in new directions. In our long history (of over over three years now, which is long in this business) we have moved from a subscription circulation, to a controlled circulation, to a requested circulation magazine. For any reader not conversant with what those magazine publishing "buzz words" mean, I shall try to explain.

Subscription circulation

Subscription circulation is delivered through the mail to a mailing list. Our mailing list largely disappeared with our separation from TPUG. Those persons who paid directly for their subscriptions will continue to get The TORPET in the mail. Subscription circulation will become a less significant part of our circulation but we will continue to welcome new readers who only have access to The TORPET by this method.

Controlled Circulation

The TORPET then turned to controlled circulation, which is distribution through designated outlets. Many large and successful magazines are distributed on that basis. Except for **The Plain Truth** most rely upon advertising for their existence. The problem is to maintain good and effective distribution outlets. Our outlets again disappeared.

What happened is that at the time we planned to enter this form of distribution we were told there were over 1,250 Commodore dealers in Canada. It turned out that nearly half of these were large chain outlets (Canadian Tire, Eatons, etc.). We ended up preparing mailing labels in November 1983 for 650 independent dealers. As Commodore began to rely more heavily on mass merchandisers even these began to fall away like flies so that as of today there are less than one hundred independent Commodore dealers still active and their numbers continue to dimminish. Once again, The TORPET had to find new readers and so now we have turned toward what appears to be a new and exciting direction.

Requested Circulation

The TORPET, like many other magazines, (such as those received by doctors, and many other professionals), will go to special groups that request it. If you are a Commodore club or a school that uses Commodores you may request copies for your club members or the computer class students in your school. We welcome requests at this time but we regret to inform you that we may not be able to honor all requests. Requests must be on school letterhead, (or in the case of clubs you must be on a published list of clubs). Please include the number of students in your classes or the average size of attendance at your club meetings.

While we wish to have as many TORPET readers as we possibly can we are restrained in the quantity that we can publish by the advertising space that we have.

In this new adventure we hope to retain many of our old friends and we expect to make many new ones. The TORPET has changed, and will continue to change. Many new readers even wonder where our name comes from (the TOR comes from TORonto which is but a small part of our circulation now) and PET was a computer made by Commodore, (many new Commodore owners have never heard of the PET).

So, hang in there, friends. The adventure continues. Whither goes Commodore, thither go we also (and perhaps also elsewhere).

EARLY NOTES ON THE COMMODORE 264

by JIM BUTTERFIELD TORONTO ON

I think it's hard for experienced Commodore 64 users to judge the 264 fairly. We've probably forgotten how hard it was to be a beginner, and so we may miss some of the most significant advantages of the 264.

The Commodore 264 is Commodore's newly-announced home computer. It's not yet available (informed guesses say late 1984), but it has been demonstrated in the USA, Canada, and Europe. Much of what I say about it is tentative; Commodore could change the design of the machine before it becomes commerically available. Some pundits wonder if the 264 will be withdrawn entirely; for my part, I believe that it will certainly be along, since Commodore has done a great deal of work on chip design and software preparation.

The thing that beginners will appreciate about the new machine is the complete dropping of technical jargon. What technical jargon? POKEs. Do you recall the time when you acquired your Commodore 64, breezing through the first few chapters of the User Guide, and then running into a brick wall? Most people bog down somewhere between, "How to make the computer beep with only 9 POKE statements" and "How to make a sprite appear on the screen with only 69 POKE statements".

All POKE statements - and their mysterious addresses - disappear in the new Commodore 264. To make the computer go beep, use two commands: VOL to set the volume, and SOUND to make the sound. The 264 doesn't have sprites, but you can predraw a high resolution graphics shape and place it anywhere on the screen (in BASIC!) with commands such as SSHAPE. Even for border and background color, you don't need to remember an address such as 53281; just give the command COLOR.

I think some 64 experts are secretly a little miffed because they have put in all that trouble learning addresses ... which won't be needed on the new 264. "No compatibility!", they cry, "It's not compatible unless programming is at least as much work on the previous machine!".

Let's make it clear that the 264 doesn't replace or obsolete the Commodore 64. The 64 has features that are not matched by the new machine, such as high quality ADSR sound and sprites. The 264 drops these "entertainment" features in favor of a significantly expanded BASIC, more screen colors, and a more advanced architecture.

Physically, the 264 is stylish and attractive. Architecturally, it has a more sophisticated structure than the Commodore 64, so that BASIC is able to power up with 60,671 bytes free even though the 264 has only the same 64K of RAM as in the C64. From a BASIC standpoint, there's a super BASIC with 4.0 disk commands, graphics capabilities, editing and renumbering features, structured statements, error trapping, IF .. THEN .. ELSE, and PRINT USING ... and more.

The keyboard shows some useful user features. The cursor keys are arranged in a diamond formation to allow the cursor to be moved as easily as with a joystick. The function keys are "preset" the moment power is on, and may be redefined as desired. Key F3, for example, is DIRECTORY when you switch on, but could changed to HELLO by typing in a simple KEY command.

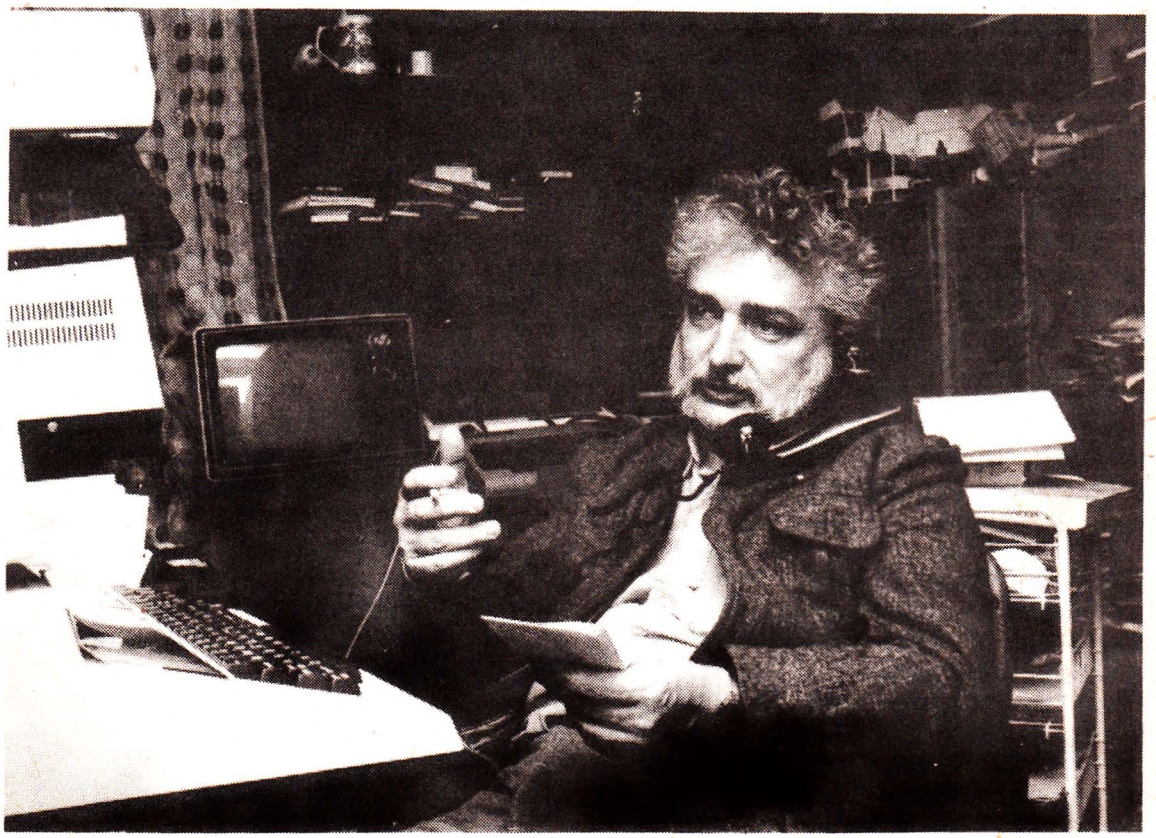
Commodore has had seven or eight software packages ready for demonstration at shows: word processors, spread sheets, LOGO, PILOT, graphics langauges and more. Programs can be plugged in on a cartridge (not the same as the 64 cartridge), or can be pre-fitted within the computer.

Commodore calls the 264 their productivity" machine: with its expanded BASIC, built-in software, and user convenience features, it will be a handy machine for getting a task done quickly. I think it's also a good learning computer. For fun, I prefer the 64 with its marvellous ADSR (attack-decay-sustain-release) sound and entertaining sprites. To get a data processing job done, I'd reach for a 264; this includes business graphics such as drawing pie charts (yes, you can do it easily in 264 BASIC, in up to 16 colors and 8 levels of brightness).

The 264 has good communications features, too. It uses a communications chip (ACIA) rather than the VIC-20 and C-64's somewhat limiting interrupt system.

Compatibility with the 64? Same 40 by 25 screen, including the same high resolution size; upwardly compatible BASIC, but POKE statements will need to be changed; same machine language, but you'll probably want to adjust the addresses. In other words, not unfamiliar to 64 users, but not totally compatible on all existing programs.

The 264 is a good machine for beginners. It's also good for programmers with "serious" applications, since effective programs can be produced quickly. For fun, for fooling around, for entertainment - I'd pick the 64. But if someone told me, "I'd like to see that report by Wednesday": I'd reach for a 264.



JIM BUTTERFIELD

NEW DIRECTIONS FOR COMMODORE

by PAUL TRACHSLER
FLESHERTON ON

NEWS FLASH!!
Commodore buys IBM to market the new CBM jr.

Farfetched? Yes, but Commodore's announcements have been creating waves around the world recently.

Commodore has a record of announcing products which never see the light of a retailer's shelves. However, two of Commodore's recent projections may be more than a designer's whim.

Let me digress for a moment and recount some of Commodore's announcements which didn't make it into the consumers' hands, or at least haven't yet. Remember the upgrade for the VIC-20 which gave it 48K of RAM built in? How about the Max, the game machine that played all the cartridges the Commodore 64 did; an Apple II emulator for the Commodore 64; the Commodore 64 with built-in two-inch color monitor (and I don't mean the SX-64); dual disk drives based on the 1541 drive format? And the list goes on. (Where is my Commodore Robot? -ed.) I've only touched on a few of the more recent "no shows", and have not given Commodore credit for the marvellous machines and tremendous advances they have made, such as the VIC-20, Commodore SX 64 and the soon-to-be-released (we think) 264 series.

In January of this year, Commodore, with much fanfare, unveiled the 264 and 364 computers at the Consumer Electronics Show in Las Vegas. These products, billed as "productivity" machines, were designed to fill the niche between the games machines and the all-out business machines; they were expected to be on retailers' shelves by April, 1984.

January 31, 1984, three weeks after Jack Tramiel's resignation, it looked like Commodore was going through an internal re-organization. After having recorded its biggest year ever in sales its share price had dropped to $36.00 after a high of $60.00 in June 1983. The 264 and 364 series were indefinitely postponed at this time; the reason was given that the Commodore 64 was still selling well and the high demand made by the 64 strained its manufacturing capability.

In February, Commodore signed agreements with two Canadian companies: Bytec-Comterm and Trillium Telephone Systems. Bytec-Comterm manufactures the Hyperion, an IBM-compatible machine, and Trillium's "TalkTo" technology expands the usefulness of the telephone system. We'll expand on these developments a little later.

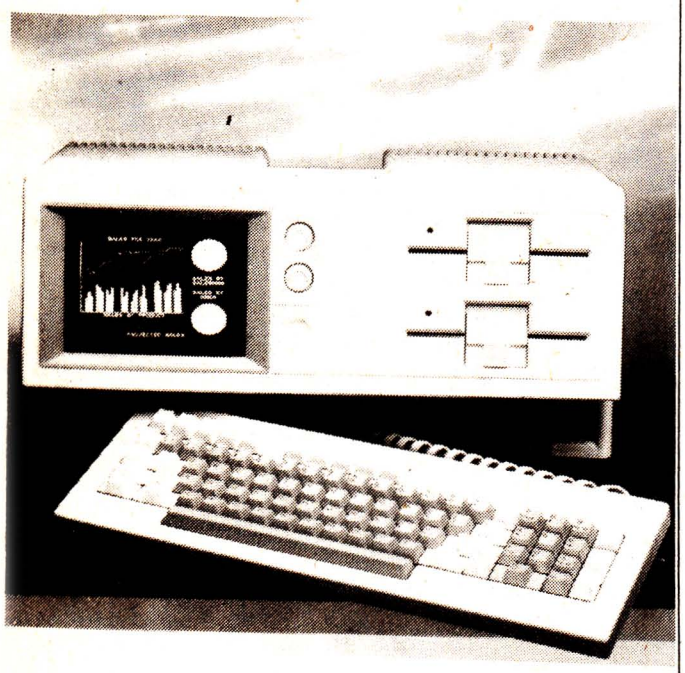
At the Hanover Fair in West Germany, Commodore revealed two new business machines, one based on the Zilog Z-8000 series CPU, which utilizes the UNIX operating system, the other using an 8088 CPU which is IBM-compatible. Commodore also announced the Commodore 16, a 16K RAM machine which may be compatible with the 264 series. The shipping date for the 264 was moved to the fall of 1984.

If your head isn't just spinning by now you are probably asking, "What does all this mean?" Well, it looks like the 264 computer **will** make it to the market, perhaps in time for the Christmas buying season. The Commodore 16 may be destined for the same resting place as the Max -- a good idea but just not economically feasible.

As to the agreement between Bytec-Comterm and Commodore, sources at both

companies are refusing comment and are attempting to play down the deal as just a sharing of information. Trillium representatives were more cooperative; however, again did not have much information to contribute, as they felt that, while newspaper reports suggested Commodore would soon be implementing the "TalkTo" technology in their new computers, they had received no word of this from Commodore.

On the upper end of the scale of computers which Commodore makes they are losing ground daily to IBM in the business PC field, as IBM sweeps over all challengers. Commodore is not likely to gain ground by introducing a new machine which is not IBM-compatible. Of the two new machines introduced at the recent fair in Hanover, West Germany, the one utilizing the UNIX operating system, may be seen as a possible smoke screen covering the actual development of an IBM clone. The IBM-compatible PC shown in Hanover by Commodore was actually a Hyperion, and this was confirmed by a source at Bytec-Comterm; however, Commodore's Press Release of April 4, 1984, makes no mention of this fact.



The Hyperion computer which is rumoured to be Commodore's challenge to the IBM PC.

Commodore also recently signed an agreement with Intel Corp., the computer chip maker, to begin manufacturing Intel's 8088 chip which is used in Hyperion, and other IBM clones, which again suggests the direction Commodore may actually be heading.

Can Commodore compete with IBM on its own turf? Analysts suggest that, with Commodore's manufacturing capability and an aggressive sales campaign, it will precipitate a general shakeout in this market, leaving the field open to Commodore and IBM. Presuming Commodore's IBM-compatible machine is very similar to, if not an exact copy of the Hyperion, we can expect an IBM PC clone including built-in monitor and dual floppies, with the added advantage of being a portable. If Commodore is able to bring this machine to market with a price tag of between $2000 and $3000, they should have a winner, with IBM ending up with some real competition for a change. Delivery dates for this and the UNIX-based machine are slated for mid 1985.

Finally, while not related to the above-mentioned machines, the Trillium deal signals a new direction for Commodore as well. Home computers are growing up and becoming more and more "useful". One area of usefulness, while always having been a dream of future gazers, has been the computer-controlled home. While interfacing with the outside world has always been "just around the corner", Commodore may be turning that corner shortly if their marriage with Trillium proves fruitful.

Trillium Telephone Systems manufactures the "TalkTo 109", a system which turns an ordinary telephone into a flexible control and communication centre. Utilizing different modules and system configurations, the "TalkTo 109" may be used as a nine-station intercom; a full-service telephone system with conferencing

capabilities, speed dial and nine 16-digit number memory, etc.- and as an energy management control system which can control lights, appliances, heating and/or cooling devices. Sounds like a great toy, doesn't it, but where does Commodore fit in?

Commodore's plans may involve combining their computer technology with Trillium's telephone expertise to come up with a new product having the best of both technologies. The dreamed-of computer home may be no further away than a Commodore 64 and "TalkTo 109" system -- a sophisticated combination of communications, energy management, burglar alarms and other remote sensing devices programmed and controlled by a home computer. One of Trillium's latest advertising brochures uses a Commodore 64 to illustrate the system's computer interfacing capability with a footnote, "Available soon".

The bottom line is that Commodore, as a company, always plays its cards close to its chest. It is constantly announcing products which never get past the prototype stage, and can always be counted on to do the unexpected. The recent developments in the company, as outlined here, only permit one to speculate on the foreseeable outcome. Keeping in mind that Commodore's strength in the past seems to have been in anticipating where everyone else will be going and getting there first, the only safe prediction may be to "wait and see". However, the next little while should prove very interesting to Commodore watchers and users and if you are in that category, the words to watch and listen for at the moment are "Bytec-Comterm", "Trillium", and "264". *

THE 264

ARE WE GOING TO GET IT?
DO WE WANT IT?
DO WE NEED IT?

by BRUCE BEACH
HORNINGS MILLS ON

This issue contains two articles about supposedly forth-coming new Commodore products. The one by Paul Trachsler describes some of the new products that have been either shown by Commodore at the Hannover Fair or have been receiving some attention in the media as Commodore prepares to challenge IBM. The other article, by Jim Butterfield, describes the 264. Some readers may well be asking, "Just what does this mean to me?"

If you, like myself, never got around to getting a SuperPET, or a B-Series, you may wonder if the world is leaving you behind. Many new owners who have just gotten a VIC of a C-64 may wonder if they should have waited.

My answer today, to anyone who doesn't have a computer, is always, "Don't wait." Computers have always improved and come down in price and there is no reason at the moment to think the trend is not going to continue. But, would you have recommended to a person in 1940 to not buy a car but to wait until 1950 because the models were going to be better. If one needed and could afford an automobile they should not have kept walking for ten years waiting for the price/performance to improve.

Because you have one computer today, it does not mean you will not want to get another later on. On the other hand one need not feel any compulsion to upgrade everytime a new model comes along. We have 13 Commodore Computers here, but we don't ever hope to have all the models, and certainly not all the ROM sets.

So, will we ever get a 264? I really don't know. In the first place i do not feel as strongly as Jim Butterfield, that we will ever see it in large quantity. Secondly, it depends on how many other people want it. If a lot of people don't want it there is not going to be a lot of software developed for it and if there is not a lot of software developed for it there are not going to be a lot of people who want it. That is the Catch-22 of all new computer systems.

There are so many C-64's out there, and there are going to be an awfully lot more with Commodore now making over 5,000 per day, that for a long time there could never be a market for 264 software products like there is for 64 software products. It seems to me that the main competitor to the 264 would be the C-64. In other words, since the demise of the TI, TIMEX, (and effectively the Adam) computers, the only real competitor to Commodore at the moment would be themselves in the low mass market end.

At one time Radio Shack had an advantage with its established dealer network. Now it is locked into its relatively fixed number of outlets while Commodore can continue to expand through the general mass merchandisers (although there may troubles develop there also as they did with the independent dealers).

The MacIntosh and PC Jr, are not real competitors to the 64. Anyone comparing prices quickly understands why. As described in the article by Paul Trachsler, Commodore may be about to challenge Big Blue, but there is no indication at the moment the challenge is going to come from the other way around. So once again, Commodore can only be creating its own competition with the 264.

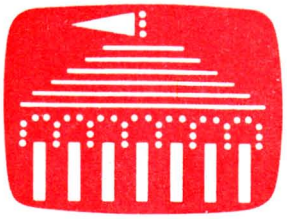
Although, the currently announced price of the 264 is higher, than what the 64 is being discounted at, it does not mean the 264 might not eventually be sold at a similar price. Given that and a lot of software similar to what a C-64 has available (and that is a BIG assumption although software publishers could quickly convert many of their programs if there was the demand) then a new buyer who does not have a C-64, and who is less acquainted with computers than the current breed of enthusiasts, might be better off getting the 264. I still only say might because one of the biggest advantages of buying a C-64 is the finding of so much support from clubs and the other many users. That would need to all be present also.

Anyway, in answer to the questions. I am doubtful we will ever see the 264 in any quantity. At the moment I really don't want one (and it is going to take a lot more sales talk to convince me differently). And I really doubt that I will ever need it.

The other newly announced Commodore machines are a different matter. They compete in a completely different price market. I have never felt that the MacIntosh and IBM products were competitors to the C-64 and VIC, but there may be forthcoming new Commodore products that will be competitors to them.

IF YOU HURRY YOU CAN HAVE IT HOME BEFORE ITS OBSOLETE!

■ Who has the best system for me? ■ Is there a specific system for my needs? ■ Can a computer reduce my costs? ■ What's the value of a network system? ■ Is it expandable? ■ What is the computer job market like? ■ How do I care for a computer? ■ What makes Micro-computers unique? ■ Will it save me time? ■ Do I have to learn to program? ■ What peripherals will I need? ■ Can I use my computer for more than one task? ■ Is it portable? ■ What about word processing? ■ Will it make my office more efficient? ■ What does the future hold? ■ What are CP/M, UNIX, MS-DOS, and XENIX? ■ Will it link up with other systems? ■ What programs will I be able to use? ■ Can I finance my computer? ■ Will it require special training? ■ Can I use a computer for financial models? ■ Are there courses I can take? ■ Will it run a spread sheet program? ■ Will computers help my budgeting? ■ Will it become outdated too quickly?



ALL THE ANSWERS ARE AT COMPUTER FAIR

Hundreds of exhibits featuring virtually every brand of micro hardware marketed in Canada today and some not yet available. See and try out the latest in Micro-Computer Technology. Software for every conceivable application - from investment programs to arcade games, music to machine language, teaching systems to accounting packages, word processors to graphic processors.

Sample hundreds of different peripherals, including printers, hard discs, modems, memory cards, video displays and synthesizers, plus accessories, publications and support services.

Daily seminars will show you how to squeeze absolutely the most power, versatility and usefulness out of your computer no matter what the application. Seminars to teach you the ins and outs of buying computers and software intelligently. Learn about new programming languages, important applications for telecommunications, what the future holds, exciting ways to use your micro-computer and more.

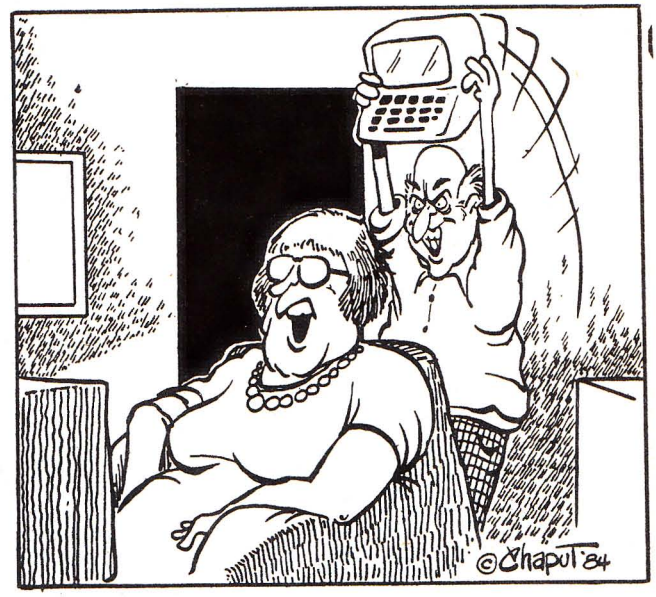
No matter what you do (or want to do) with your computer the Computer Fair seminars will help you do it better and best of all, the cost of the seminars is included with your admission fee.

For more information contact: Hunter Nichols Inc., 721 Progress Ave., Scarborough, Ontario. Tel: (416) 439-4140.

COMPUTER FAIR

May 10-13
International Centre Toronto

Hunter Nichols Inc.



Q.K. FRANK, SHOW ME ONE THING YOUR COMPUTER IS GOOD FOR!

SPEECH SYNTHESIS FOR THE COMMODORE 64

Gregory L. Halley, M.D.,
Silver Spring MD.

Have you been wondering when you would finally have the capability of providing your Commodore 64 computer with the gift of speech. Well, Tronix has recently released the Commodore 64 version of its Software Automatic Mouth (S.A.M.) disk based speech system. (1) This popular software package has already been available for the Apple, Atari and VIC-20 for some time now. (2)

The system itself is very easy to access from the resident BASIC provided with the Commodore 64 computer by using a software "WEDGE". The software wedge provides you with the option of using only phoneme based speech or using a direct English-Speech translation. The format for using the direct English-Speech command from Commodore 64 BASIC can be accessed by direct quote or string variable as demonstrated below:

```
10 A$="during these turbulent times"
20 ]SAY "now is the time for..."
30 ]SAY A$
40 Goto 100
```

It is also noteworthy that you have a remarkable amount of control over the pitch, stress and speed of speech. This control is obtained by using the BASIC commands of]SPEED,]PITCH and]KNOBS from within the actual BASIC program utilizing the speech synthesis capability. The format for using these commands is outlined below:

```
10 ]RECITER
20 ]SPEED 60
30 ]PITCH 55
40 ]KNOBS 125,176
50 ]SAY "This is a trial speech pattern"
```

EXPLANATION OF BASIC COMMANDS

Line 10 - Calls the Reciter machine language subroutine which allows the program to use direct English-Speech translation.

Line 20 - The Speed command sets the speed of speech production by the Reciter subroutine. The range of values for Speed is 0 - 225 where the speed is inversely proportional to the value (i.e. 225 is very slow and 20 is quite fast).

Line 30 - The Pitch selection determines the quality of the spoken voice. The range of values for Pitch are 0 - 255 where the highest value corresponds with the lowest voice and the lowest value produces the highest voice.

Line 40 - The Knobs command essentially provides a means of changing the quality of the voice without affecting the Speed or Pitch of the voice. The analogy used in the user guide compares the Knobs control to defining the size of the mouth and the throat. The range of values is 0 - 255 with a direct correspondence between the higher values and a larger throat and mouth (i.e. 20 is a small mouth whereas 225 is a large mouth).

The Reciter subroutine called from BASIC provides a convenient means of providing English-Speech translation of material which varies from one run to the next. However, the system also includes a mechanism for providing a more refined control over speech which may be repetitive in nature (i.e. in a program which utilizes a small speech vocabulary). This system is based upon the conversion of

English words into spoken phonemes. These phonemes are then placed within the Speech string in place of the English words and spoken with a remarkable degree of clarity when compared with the direct English-Speech translation. In addition, this phenome based system allows direct control of stress points within the program. This is done by placing a numeric value of 1 - 8 directly after the vowel to be stressed. This is useful because of the importance of context in spoken English and the ability this feature gives the programmer in conveying context through stressing key words, phrases, etc.

This all sounds fine and good you say, but what about the price you will pay in loss of BASIC RAM? Well, the truth is that the S.A.M. package can be very frugal if used properly. The machine subroutine which drives S.A.M. is approximately 10.75 K in length but the noticeable decrease in BASIC RAM is only approximately 2.75 K. This is possible because most of the S.A.M. package is located in memory not accessed by the BASIC operating system. If you decide to use the Reciter subroutine which allows direct English-Speech synthesis you will use additional RAM; however, this will conflict with the DOS WEDGE. The alternative placement of Reciter into low memory will consume an additional 6.0 K RAM.

In conclusion, the advantages of the S.A.M. system can be summarized as: 1) Relatively inexpensive (sugg. retail ($60.00) 2) No additional hardware and resulting loss of user/expansion port as required in other speech systems 3) Drive software provided to provide direct English-Speech translation at no additional cost as is true with some other systems 4) Inclusion of a phoneme dictionary with the User manual provided with the software package 5) It is available now not at some undefined point in the future that never seems to arrive. The disadvantages might be summarized as 1) Quality of speech dependent upon the quality of the monitor speaker 2) Loss of BASIC RAM in programs which may require a large amount of RAM 3) "Machine sounding" voice, the system not able to provide the more human-sounding voices as can other systems -- but, then again, it doesn't have their price either.

Letters on inquiry for the product may be directed to:

TRONIX
8295 S. La Cienega Blvd.
Inglewood, CA 90301
U.S.A.

(1) S.A.M. registered Trandemark of TRONIX, Inc.

(2) APPLE registered Trademark of APPLE Computer
ATARI registered Trademark of ATARI Computer
VIC-20 is registered Trademark of COMMODORE ELECTRONICS, LTD. *

Garbyte by DAN SLOAN

VOTRAX BASED SPEECH SYNTHESIZER

by V. SIRUGO
SCARBOROUGH ON

Your VIC-20 can be given a more human like quality if you attach a Speech Synthesizer to it. For the pre-schooler a Speech Synthesizer can turn your computer into a friend. I recently set out to find a Speech Synthesizer for my VIC 20.

After spending the better part of an afternoon calling all Commodore suppliers in the Toronto area without luck, I decided to call Commodore themselves (Toronto Office). Their response was simply "they think it is still in the development stage, but they weren't sure". If you want any further information then contact the U.S. office. "A third party vendor" was now my only alternative.

I located two Speech Synthesizers for the VIC-20. Two more have since shown up on the market. The first from PROTECTO - a company operating out of Barrington, Illinois, the second being the SPEAKEASY from PERSONAL PERIPHERALS - operating out of Aurora, Illinois. Based on their advertisements both of these synthesizers seemed to be similar. I ended up choosing the Speakeasy because the advertisement said you could connect it to your TV set if desired. There was no mention of this from PROTECTO, although I now do believe that this may be possible with theirs too. Now to describe the Speakeasy.

The Speakeasy is a votrax based speech synthesizer containing a VOTRAX SC-01 integrated circuit. This circuit allows the Speakeasy to pronounce any of the 64 votrax phoneme. Each phoneme can be pronounced at any of four different inflection levels. Phonemes are pronounced together thus creating a word.

An example should show this more clearly. The word "ONE" is made of 3 votrax phonemes, these being "W" as in WATER, "UH" as in UP, and "N" as in NEW. All of these three symbols has a corresponding number. "W-45, UH-51, N--13". If you want the Speakeasy to pronounce the word "ONE" then you must first POKE each one of these values into location 38912 of the VIC 20. A short program that does just that follows.

```
10 READ A
20 POKE 38912,A
30 IF PEEK(38912) < 128 then go to 30
40 IF A = 03 then STOP
50 GO TO 10
60 DATA 45,51,13,03
```

Line 10 makes use of the READ statement to get each one of the numbers corresponding to the votrax phonemes.

Line 20 POKES the number into location 38912.

Line 30 - When the Speakeasy has finished pronouncing the sound corresponding to the number that you have entered, bit 7 in location 38912 will change to a 1. This change may be detected by checking the location and seeing when it becomes greater or equal to 128. Thus you have to remain in a loop until this happens.

Line 40 - the value 03 is not really a sound and has nothing to do with the direct pronunciation of the word "ONE". It does though cause the last sound pronounced to decay slowly, thus resulting in a better pronunciation of the word. We use it here to not only pronounce the "ONE" more legibly but to also signal the end of the word.

Line 50 merely sends us back to the beginning of the program in order to read

in the next phoneme number.

Line 60 is a data statement containing our votrax phoneme numbers corresponding to the word "ONE".

Each one of the Speakeasy sounds can be pronounced in any of four different inflection levels in order to emphasize a particular syllable, or the entire word. To accomplish this you add 0 to the phoneme number for inflection level 0, 64 for inflection level 1, 128 for inflection level 2, and 192 for inflection level 3.

The Speakeasy itself looks similar to game cartridges and plugs into the VIC 20 expansion port. Two screws are accessible through the cartridge via two small holes. One of these screws controls the volume for the Speakeasy The other controls the pitch. On the back of the cartridge to a speaker via this RCA plug. I have connected the unit to my TV by attaching wire (via a small alligator clip) to the RF modulator of the VIC and connecting it to the Speakeasy Full instructions for even the novice (which I am) are included with the Speakeasy

Finally the Speakeasy comes with a 7 page instruction manual which explains everything (I've said here). The instruction manual also includes a table of all 64 votrax symbols. These symbols correspond very closely to the symbols used in most standard dictionaries. Thus a dictionary, which breaks up a word for pronunciation purposes can be most helpful. Finally the instruction manual contains a list of approximately 250 words already broken down into their votrax symbols. It is up to you though to translate these symbols into their corresponding numbers. This job is not hard, just tedious. Since I have already done this I have included at the end of this review the entire dictionary and their votrax phoneme numbers.

The Speakeasy also includes two programs. 1) A phoneme editor and 2) a phoneme speaker to aid you in the creation of words, or sentences. The editor displays all 64 phoneme symbols for you on the screen and by means of the keyboard allows you to copy any of the symbols to a buffer area at the bottom of the screen, translating the symbols to numbers for you on the way. On command it will pronounce any of the symbols for you and will also pronounce the entire buffer that you have created. The buffer can then be saved, as a data file to tape or disk.

The speaker program reads in a data file of phoneme numbers previously created by the editor and pronounces each word for you.

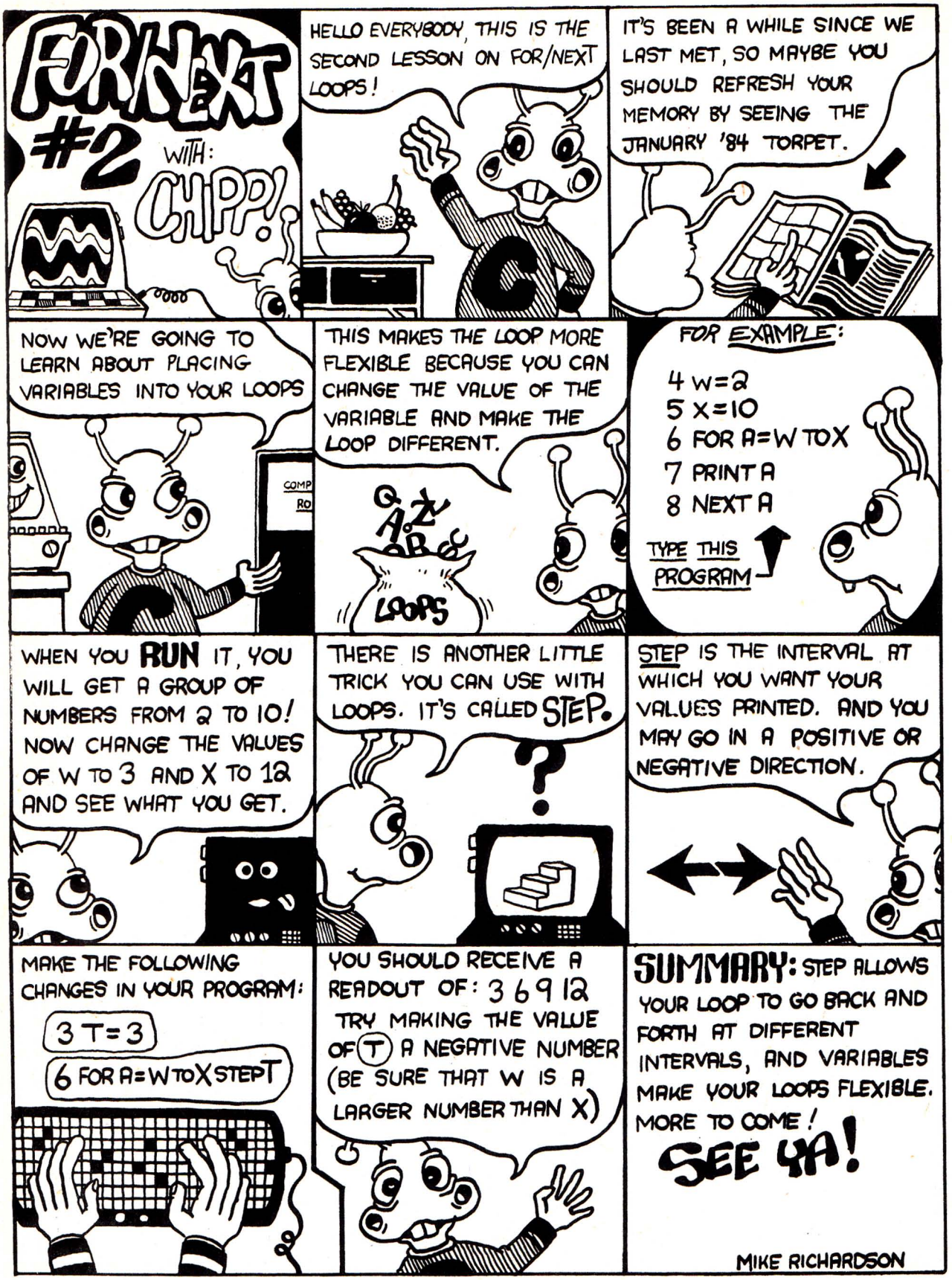
All in all the Speakeasy is easy to use, although somewhat cumbersome at times. It works on an unexpanded VIC and can make that program you've been working on slightly more friendly.

A final note. I have just learned that a machine language program called SMOOTHTALKER is now available from Personal Peripherals. This program sells for $19.95 (US) and requires an 8K expansion that can be placed into block 5. It is designed to work with the Speakeasy from within BASIC. It adds a new BASIC command called SPEAK which works similarly to the PRINT command. That's about all I know about it for now. When I acquire one I'll let you know all about it.

The Speakeasy Cartridge, Instructions & Dictionary, Cassette Editor, and Extension Speaker is available for $59.00 (US).

SAMPLE OF SPEAKEASY VOCABULARY

A	6 33 41
B	14 14 44 41
C	90 90 44 41
...	
Y	45 21 10 41
Z	18 18 44 41
1	45 51 13
2	42 54 55 55
3	57 43 60 41
...	
100	27 50 13 30 43 01 04
1,000	57 21 35 55 18 00 13 30
Monday	12 50 13 30 06 33
Tuesday	42 54 55 18 30 06 33


FOR/NEXT #2 WITH: CHIPP!
HELLO EVERYBODY, THIS IS THE SECOND LESSON ON FOR/NEXT LOOPS!
IT'S BEEN A WHILE SINCE WE LAST MET, SO MAYBE YOU SHOULD REFRESH YOUR MEMORY BY SEEING THE JANUARY '84 TORPET.
NOW WE'RE GOING TO LEARN ABOUT PLACING VARIABLES INTO YOUR LOOPS
COMP RO
THIS MAKES THE LOOP MORE FLEXIBLE BECAUSE YOU CAN CHANGE THE VALUE OF THE VARIABLE AND MAKE THE LOOP DIFFERENT.
LOOPS
FOR EXAMPLE:
4 W=2
5 X=10
6 FOR A=W TO X
7 PRINT A
8 NEXT A
TYPE THIS PROGRAM
WHEN YOU RUN IT, YOU WILL GET A GROUP OF NUMBERS FROM 2 TO 10! NOW CHANGE THE VALUES OF W TO 3 AND X TO 12 AND SEE WHAT YOU GET.
THERE IS ANOTHER LITTLE TRICK YOU CAN USE WITH LOOPS. IT'S CALLED STEP.
STEP IS THE INTERVAL AT WHICH YOU WANT YOUR VALUES PRINTED. AND YOU MAY GO IN A POSITIVE OR NEGATIVE DIRECTION.
MAKE THE FOLLOWING CHANGES IN YOUR PROGRAM:
3 T=3
6 FOR A=W TO X STEP T
YOU SHOULD RECEIVE A READOUT OF: 3 6 9 12
TRY MAKING THE VALUE OF (T) A NEGATIVE NUMBER (BE SURE THAT W IS A LARGER NUMBER THAN X)
SUMMARY: STEP ALLOWS YOUR LOOP TO GO BACK AND FORTH AT DIFFERENT INTERVALS, AND VARIABLES MAKE YOUR LOOPS FLEXIBLE.
MORE TO COME!
SEE YA!
MIKE RICHARDSON

If It Worked Once It'll Work Again And Again And Again.....

Larry Goldstein, Bolton, ON.

The operations included in this article are sufficient to design some very useful and attractive routines. Let your imagination run loose with them.

A program which needs a separate command for every operation is tedious to write and slow to run. Filling a screen, for example, would require between 500 (for a VIC) and 2000 (for an 8032) LDA and STA commands to put the information where you want it. In BASIC we would use a FOR...NEXT loop to accomplish this; in Machine or Assembly Language we use branching loops.

Consider the following BASIC program (substitute the appropriate values for your machine for the stars in line 10):

```
5 FOR V =38400 to 38905:POKE V,1:NEXT:REM FOR VIC ONLY
ONLY
10 SM = ****: REM USE 32767 FOR PET/CBM, 7679 FOR
VIC, 1023 FOR C-64
20 FOR I =  1 to 19
30 READ Q
40 POKE SM + I, Q
50 NEXT
60 DATA 19,21,2,19,3,18,9,2,5,32
70 DATA 20,15,32,20,15,18,16,5,20
```

Obviously, this routine will read the data one at a time and place them in screen memory starting at the upper left hand corner of the screen and ending 18 spaces to the right. In Assembler we would LOAD the data into the Accumulator and then STORE them in screen memory, using the X- or Y-Register as a counter or index.

To duplicate this programme in Assembly Language, load Supermon into your machine, and RUN it, then enter the following:

```
A 034B LDX #$13
A 034D LDA $0356,X
A 0350 STA $7FFF,X
A 0353 DEX
A 0354 BNE $034D
A 0356 RTS
```

*** For VIC substitute $1DFF in the third line. For C-64 use $03FF.
("A" is the instruction to perform an assembly.)

The first line is pretty well self-explanatory. We simply load the number $13 (hex) or 19 (decimal) into the X-Register to act as our counter.

Next we load the accumulator with the contents of memory location $0356+X ($0356 + $13 = $0369).

Then store this value in memory location $7FFF+X or $8012 (for PET/CBM). This location is part of screen memory, so the character appears on the screen.

Now decrease (or decrement) the contents of the X-Register by 1, resulting in a value of $12. Whenever a numerical transaction like this occurs, there may be a change in the Status Register. If the result of the transaction is zero, this is noted, as is a negative result. The result, $12, would be noted as being non-zero and non-negative. The next operation uses this information.

BNE means "Branch if the result is Not Equal to zero". In Assembler, this command is followed by the memory location that you want the programme to go to. This instruction resets the Programme Counter to this address, and the branch is accomplished. However, the branch can only be a maximum of 128 steps forward or backward.

The programme now branches back to the LDA command with X set at $12, so another datum is loaded and stored in another screen location. This procedure operates in reverse compared to the BASIC version, starting at the end of the message and working its way back. Eventually the

X-Register contents will become zero and the branch will not work (the last datum used will have been loaded from $0356+1) so the programme will proceed to... RTS which returns the computer to BASIC (in this case).

So far we have no data to work with. Let's get out of the assembly mode by pressing RETURN. Since the programme has taken us to location $0356, we can store data starting at $0357 (which is why the address $0356 was chosen in the second line of the routine). Enter the instruction, M 0357 0367, and you will see a display of a number of memory locations starting at $0357. Simply change the two digit numbers to duplicate the ones below by typing the changes right over the display and pressing RETURN at the end of each line. (The VIC display will look a little different because of the shorter line lengths, but the principle is the same.)

```
0357 13 15 02 13 03 12 09 02
035F 05 20 14 0F 20 14 0F 12
0367 10 05 14 00 00 00 00 00
```

These are the same numbers as in the BASIC DATA statements but in hexadecimal notation. The zeroes at the end mean nothing. Exit Supermon by entering X RETURN, clear the screen, and RUN the routine with SYS 843. (VIC NOTE: When you clear the VIC screen you also clear colour memory and your display is invisible. Use line 5 from the BASIC programme to fill colour memory, then SYS 843.)

If you now reactivate Supermon (SYS 4 for PET/CBM, SYS 8 for VIC and C-64), you can take a fuller look at the programme. Once Supermon is running again, enter D 034B RETURN and you will get a screen full of stuff. The columns of numbers on the left side of the screen are the memory locations of the commands followed by the machine language number (pattern) for the command and the two or three number address (if any). Note that the two byte address $0356 is stored in reverse or LOBYTE/HIBYTE order. Finally, there is the Assembly Language version, which is not actually stored in memory, but has just now been translated (or disassembled) by Supermon for our convenience.

We have begun to accumulate quite a repertoire of commands, so let's take another look at them.

LDA, LDX and LDY are commands to put a value into the Accumulator or the X- or Y-Register. We can specify the actual value to be loaded as we did with LDX #$13. This is called Immediate Mode addressing and works for all three registers. This method of addressing is easy to use and to understand, but it is not very flexible, since the actual value must be provided each time the command is used. We could also specify "load the accumulator with the value found at memory location $1234" using the command, LDA $1234, and similarly LDX $1234 or LDY $1234. In giving the specific address from which the value is to be fetched, we are using Absolute addressing. Finally, we specified "load the accumulator with the value to be found in memory X steps away from location $0356". This method is very handy when wanting to use a series of values, as we did here, or one of a series of values depending on the value of the index. This is Absolute, Indexed addressing, and we could use either the X- or Y-Register value as the index.

STA, STX and STY operations are also available with Absolute and Absolute, Indexed addressing modes.

The decrement commands, DEX and DEY decrease the value in the respective registers by 1. If the register holds a zero, then DEX or DEY yields a value of 255 ($FF). It's just like rolling back a hexa`-decimal odometer. The opposite commands, INX and INY, will increment the registers or increase them by 1.

Since the X- and Y-Registers can hold values only up to 255 ($FF), the indexed addressing modes can only reach 255 locations ahead of the starting address. In other words LDA $0356,X can only LOAD from $0356+FF at the maximum and STA $7FFF,X can only store in location $7FFF+255 maximum.

The branch command BNE (and its opposite BEQ or "Branch if the result is EQual to zero") respond to the Status Register and whether or not it holds in-

formation of a zero-result. BNE executes the branch if there was not a zero result, while BEQ branches if the result was zero.

In running the loop "backwards" we simplified things by specifying the extent of the loop by setting the index to $13 at the beginning and using the ability of the machine to distinguish between a zero and a non-zero. We could have done it the other way around by setting X to 1 and then using INX each time, checking to see if it had reached $14 yet, and then doing the branch if X was less than $14. This requires an extra step, checking for $14, which uses up memory and wastes time when running.

If we want to affect the whole screen using routines similar to this one, we run into the limitation mentioned previously, that indexed addressing can only affect up to 256 addresses (for X = 0 to 255). The simplest way around this is to put in a number of commands which will affect the whole screen. For a 40-column PET, screen memory starts at $8000 and includes 1000 locations. So we can fill the screen with the letter A using a routine like:

```
A 033E LDA #$01
A 0340 LDX #$00
A 0342 STA $8000,X
A 0345 STA $8100,X
A 0348 STA $8200,X
A 034B STA $8300,X
A 034E DEX
A 034F BNE $0342
A 0351 RTS
```

The third line will put character #1 (i.e. A) in locations $8000 to $80FF, the next line carries on for the next 256 locations and so on. Notice that the first execution of the routine fills $8000, $8100, $8200, and $8300. The index then becomes $FF, the branch executes and repeats until X becomes 0 again. To run the same on the 8032, we would need three more lines continuing to $8600,X. The VIC needs only two lines to fill its 506 locations ($1E00,X and $1F00,X) while the C64 would need four lines (STA $0400,X STA $0500,X STA $0600,X and STA $0700,X--don't do it yet: read on).

Another piece of memory of interest is colour memory in the VIC and 64. Every screen location is represented by a colour memory location whose contents determine the colour of the character displayed there. Using the routines shown above, but changing the addresses to colour memory instead of screen memory, allows you to change the entire display instantly. For example:

```
A 033E LDA #$02
A 0340 LDX #$00
A 0342 STA $9600,X
A 0345 STA $9700,X
A 0348 DEX
A 0349 BNE $0342
```

This routine will fill VIC colour memory with the value of two, resulting in a red display. If entered as shown it will fit in just ahead of our previous advertising message, and a SYS 830 will give a visible display without the "line 5" routine. A similar routine can be written for the 64 to change the display colour of all or part of its screen. If used by itself, the routine would end with RTS to return to BASIC when its finished.

In experimenting with the VIC remember that screen and colour memory shifts depending on whether or not you are using a memory expander. The locations are as follows:

	Unexpanded	Expanded
Screen Memory	7680 - 8191 $1E00 - $1FFF	4096 - 4607 $1000 - $11FF
Colour Memory	38400 - 38911 $9600 - $97FF	37888 - 38399 $9400 - $95FF

The screen memory of the C64 can be shifted all over the place with starting addresses ranging from 0 to 15360 ($0000 to $3C00), but the basic location is 1024 to 2047 ($0400 to $07FF). When changing screen memory, remember that the last 8 bytes at the end of screen memory (wherever it's located) are used as sprite pointers. This means that a routine including $0700,X , for example, could clobber your sprites. To be safe, use $06E0,X instead. Colour memory does not move, and can always be found at 55296 - 56295 ($D800 - $DBE7). *

MICRO-FAX

54 FLERIMAC ROAD, WEST HILL, ONTARIO M1E 4A9 CANADA
TELEPHONE: (416) 282-1532

THE 64 SOFTWARE HOUSE

ENTERTAINMENT

A.E.	"BRODERBUND"	D.	39.95
SPARE CHANGE	— " —	D	39.95
OPERATION WHIRLWIND	— " —	D	40.95
ZAXXON	"SYNAPSE"	T/D	44.95
QUASIMOTO	— " —	T/D	39.95
SHAMUS CASE II	— " —	T/D	39.95
SORCERER	(INFOCOM)	D	59.95
FLIGHT SIMULATOR	(SUBLOG)	T/D	59.95
SAMMY LIGHTFOOT	(SIERRA)	D	33.95
PAINT BRUSH	(HES)	C	23.95
BRUCE LEE	(DATA SOFT)	C/D	39.95
CASTLE WOLFENSTEIN	(MUSE)	D	33.95

BUSINESS

HOME ACCOUNTANT	(CONT)	D	76.95
VIP TERMINAL		D	65.95
ELECTRONIC CHECKBOOK	(TIMEWORKS)	T/D	27.95
BANK STREETWRITER		D	72.95

SPECIALS

KOALA PAD AND PAINTER & SLICK STICK (JOYSTICK)			109.95
WHIZ KIDS INTRO TO BASIC		D	29.95
STRIP POKER		D	37.95
STAR MAZE		D	38.95
INTRODUCING THE RITEMAN PRINTER 120 CPS 1 YR. WARRANTY			499.00

ONTARIO RESIDENTS ADD 7% SALES TAX.
ORDERING & TERMS: SEND CASHIER CHECK, MONEY ORDER, OR CERTIFIED CHECK. VISA/MASTERCARD PLEASE INCLUDE CARD NUMBER & EXPIRY DATE AND SIGNATURE. ADD $2.50 FOR SHIPPING AND HANDLING.
ALL ITEMS SUBJECT TO AVAILABILITY. PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
FOR CATALOG SEND $1.50 REDEEMABLE.

TEO™ Dot Matrix Printer

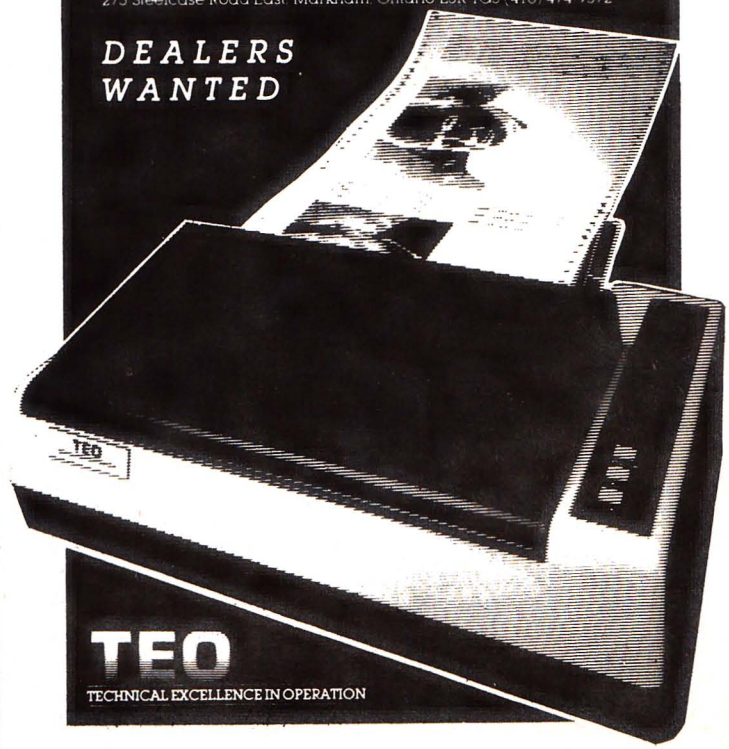
PRICE BREAKTHROUGH
$499.00

- Compatible with Commodore 4023
- Fast 80 CPS printing
- Bidirectional printing
- Dot-addressable graphics
- Many character styles and fonts
- Friction and tractor feed
- Near letter-quality printing
- Includes Vic-20 / Commodore-64 hardware interface. Works with all software.

Contact TEO Computers for the name of your nearest dealer.

TEO Computers and Peripherals, Inc.
275 Steelcase Road East Markham, Ontario L3R 1G3 (416) 474-9372

DEALERS WANTED



TEO
TECHNICAL EXCELLENCE IN OPERATION

C-64 VIC 20 ATARI

CHILD DEVELOPMENT SERIES

(for the 3.5K VIC and 16K ATARI)

BECi

ADD/SUB—$16.95
Displays single or multiple digits with or w/o pictures, borrows, carries, scoring, and audio/video feedback.

NUMER-BECi—$16.95
Number recognition, object counting, object grouping, and number/size/shape discrimination.

ALPHA-BECi—$16.95
Twenty-six screens with letters/pictures/labels 'built' on the screen

prices in U.S. dollars

Dealer Inquires Invited

BOSTON EDUCATIONAL COMPUTING, INC.
78 Dartmouth Street, Boston, MA 02116
(617) 536-5116 *MA res. add 5% tax

MasterCard VISA

TEXT TO BASIC CONVERSION QUICKLY

by PHIL KEMP
EDMONTON ALTA

Articles in TORPET and elsewhere have described ways to "tokenize" BASIC text - i.e. to take lines of text characters and produce BASIC statements in memory ready to be RUN. This note describes a fast-running program for the 64 to convert text from a disk file.

THE BASIC APPROACH

The conventional method uses a small BASIC program to read lines of text from a disk file, display them on the screen, and then simulate the pressing of the RETURN key. The lines are then tokenized and added to the BASIC program in memory just as if they had been keyed.

The method works quite well, except for one drawback. It is slow: very slow for a large program. Since the keyboard buffer holds only ten characters, it turns out that we can display and process only eight lines of text at one time. Then the BASIC program must end, so that BASIC's editor can process the displayed lines. Now, whenever BASIC's editor adds a line to the program in memory, the BASIC program location pointers are reset, and all disk files are closed. So, though we may automatically reenter our conversion code and process further batches of eight lines, each time we do so we must reopen the disk file and read through all the text already processed.

As a mathematical exercize, if our input file contains N lines of BASIC, and we process groups of eight at a time, then we must read the equivalent of N/16 times the whole file. For a 400-line program, it's like reading the whole file 25 times - no wonder it's slow!

ALTERNATIVE APPROACHES

An obvious approach is for our starter program to tokenize the input text itself. This way, the file can be read once only. A recent program on a TPUG disk does just that. Very slowly. And it would require a fair bit of extra work to cope with language extensions such as Simons BASIC.

Or, we can take advantage of the generous memory available on the 64, by reading the whole file to memory, and then process it line-by-line. Again, the file is read once only. Again also, using BASIC's editor to do the tokenizing resets BASIC's pointers, so we have to keep track of actual memory location where our text is. Not difficult, but awkward, and our program will be littered with slow-running PEEKs and POKEs.

A better approach is to keep the latter method, and to code the program in machine language. Then we have a fast-running program which does not require clean-up at the end of execution. This provides an excellent exercize in straightforward assembler programming, involving simple disk I/O operations. As an extra benefit, we are left with a set of subroutines useful for other projects. (Actually, the original version of the program to be described was thrown together one evening by recycling routines from another program).

THE ASSEMBLER PROGRAM

The program listing shows source arranged for the PAL Assembler. No significant change is needed for the CBM Assembler package. The original code was created using SUPERMON (not difficult, just tedious typing).

I've chosen to locate the code starting at hex location $C000 (just above BASIC ROM). To locate elsewhere, only statements 30 and 670 need changing.

The program is composed of several sub-routines, useful in other programs also.

USING THE PROGRAM
To start execution, we type SYS49152 (i.e. $C000). A disk file of type SEQ is then copied into memory by routine GETFIL. This uses routine FNGET to set the name of the input disk file. In case of a typing error, the DEL key provides limited editing capability.

Routine DISLIN puts the first line of text at the top of the screen, followed by the direct BASIC statement "SYS49155". The keyboard buffer is loaded with HOME and two RETURN codes, and control is passed to BASIC via the normal BASIC "warm start" routine. The BASIC editor tokenizes the displayed line, and merges it into any BASIC program already in memory. The SYS49155 (i.e. $C003) automatically gets us back into our conversion program to deal with subsequent lines.

As text lines are processed, they appear briefly at the top of the screen in turn; we can easily monitor progress and see just how fast this program runs. At the end, the message "ALL DONE" is shown, followed by the usual READY. If we wish, we may then merge further files into the BASIC program, by entering SYS49152 again.

USE OF MEMORY
The placement of the input text in memory is important. Routine BUFFINIT initializes buffer pointers 1024 locations past the current end-of-BASIC-text (VARTAB). This way, even if we have a few lines which, when tokenized, are longer than the original text, there will be no overlap between current BASIC text and the unprocessed buffer content.

There are two buffer pointers - NEXTFREE, which points to the next unused buffer location, and NEXTSHOW, which points to the next buffer character to be output. These must be located in page zero, and we cannot appropriate any of BASIC's pointers, since we are using the BASIC editor to tokenize text. The FREKZP area at $FB though $FE provides a safe place.

ERROR HANDLING
Before opening the input file, OPEN15 opens the disk error channel. After the file open attempt, the error channel is read. If either of the first two characters are not zero then a file open error is indicated. During file input, the I/O status flag is monitored for errors. A check is also made to ensure that the buffer does not extend beyond the current "highest memory used by BASIC" (MEMSIZ) location. Any error detected results in a message and a chance to specify a new input file. If no file name is given, the program ends.

SOME DETAIL POINTS
The use of the small (two element!) "jump vector" at the start of the program code enables the entry points to remain fixed in location, while the real code may move around as program development progresses.

The call to the KERNAL routine CLRCHN on statement 1700 deserves note. If this is omitted, any subsequent calls to routine GETIN result in reading characters from the error channel instead of the keyboard! The result in this case is a program loop.

Another precaution is the call to CLALL in statement 1545. When we RUN a BASIC program, some initialization of pointers and files is done before execution begins. But, when we use the SYSnnnnn statement to start a machine-code program, no initialization is done. So, the JSR CLALL is a precatuiton to ensure that there are no files left open at the start of execution.

OTHER USES
The program was designed to tokenize BASIC text. As a side benefit, it also provides a program MERGE facility.

Many of the routines can (and will) be used as building blocks for other programs. This is made easy by this conversion program itself. Like many programmers, I use the PAL Assembler package. Now, source for PAL is stored in the same form as BASIC code (and words like PRINT and DATA are tokenized, too). So, to use a routine from this program in

another, all I need do is LOAD this program source text and LIST the relevent lines to a disk file. Then I LOAD the source of the new program, and execute this conversion program to merge the routine into the new program. The only real pitfall to watch for in this manoeuvre is to ensure there are no common line numbers in the statements to be merged.

THE BOTTOM LINE

It's time to look at the costs and benefits of developing and keying this program. The price is clear enough; some 250 lines of assembler code to be keyed - either as source for Assembly, or entry via a machine language monitor. The benefits are, I believe, substantial. We have a fast conversion program, useful for BASIC (including extensions) and also for PAL source. We have a set of routines useful for other projects, and a tool to easily copy the code to a new program. Finally, we have exercized our knowledge of Assembler programming and of the 64's operation, and learned a little in the process.

```
2
20:     c000                    .opt p4
30:     c000                    *=   $c000
40:     c000 4c 5d c2           jmp  start    ; get text file, process
                                                  first line
50:     c003 4c 60 c2           jmp  start2   ;  process next line
                        ; kernal routines
70:     c006            setlfs  =    $ffba
80:     c006            setnam  =    $ffbd
90:     c006            open    =    $ffc0
100:    c006            close   =    $ffc3
110:    c006            chkin   =    $ffc6
120:    c006            chkout  =    $ffc9
130:    c006            clrchn  =    $ffcc
140:    c006            chrin   =    $ffcf
150:    c006            chrout  =    $ffd2
160:    c006            clall   =    $ffe7
170:    c006            getin   =    $ffe4
                         ;
190:    c006            rbasic  =    $a480    ; return to basic
                                                  (temporarily)
200:    c006            kbbuff  =    631      ; start of keyboard buffer
210:    c006            kbcnt   =    198      ; no. of chars in kbbuff
220:    c006            stflag  =    $90
230:    c006            vartab  =    $2d      ; pointer to end-of-basic
                                                  text
235:    c006            memsiz  =    $37      ; highest address for basic
                         ;
250:    c006            nextfree =   $fb      ; pointer - next free byte
                                                  in buffer
260:    c006            nextshow =   $fd      ; pointer - next character
                                                  to display
270:    c006 41 41 41 fname     .asc "aaaaaaaaaaaaaaa,s,"
280:    c018 01         fnwr    .byte1          ; flag for file i/o
310:    c019 00         fnlen   .byte0
```

```
330:  c01a 00        charno   .byte0
                      ;
350:  c01b 20 cc ff  closf    jsr  clrchn            ; clear channels
360:  c01e a9 04              lda  #4                ;  and close
370:  c020 20 c3 ff           jsr  close             ;  file 4
380:  c023 60                 rts
                      ;
400:  c024 0a        msgout   asl           ; output msg routine
410:  c025 aa                 tax           ; a=msg number
420:  c026 bd d6 c0           lda  msgadd,x
430:  c029 8d 37 c0           sta  ms1+1             ; set start of
440:  c02c bd d7 c0           lda  msgadd+1,x        ;  message address
450:  c02f 8d 38 c0           sta  ms1+2             ; a=msg number
460:  c032 a2 03              ldx  #3
470:  c034 86 9a              stx  $9a               ;o/p device=screen
480:  c036 ad 99 99  ms1      lda  $9999             ; get character
490:  c039 f0 0d              beq  ms2               ; end at null
500:  c03b 20 d2 ff           jsr  chrout            ; o/p character
510:  c03e ee 37 c0           inc  ms1+1             ; move up pointer
520:  c041 d0 f3              bne  ms1               ; get next
530:  c043 ee 38 c0           inc  ms1+2
540:  c046 d0 ee              bne  ms1
550:  c048 60        ms2      rts
                      ;
570:  c049 0d 00     msg0     .byte13,0
580:  c04b 20 0d 00  msg1     .byte" ",13,0
590:  c04e 4e 41 4d  msg2     .asc "name too long"
600:  c05b 0d 00              .byte13,0
610:  c05d 4f 50 45  msg3     .asc "open error"
620:  c067 0d 00              .byte13,0
630:  c069 49 2d 4f  msg4     .asc "i-o   error"
640:  c074 0d 00              .byte13,0
650:  c076 46 49 4c  msg5     .asc"file name for text to be converted"
660:  c098 0d 00              .byte13,0
670:  c09a 53 59 53  msg6     .asc "sys49155    ";  goto $c003 to
680:  c0a6 0d 00              .byte13,0           ;   process one line
690:  c0a8 0d        msg7     .byte13
690:  c0a9 52 45 4d           .asc "rem  all done"
700:  c0b6 0d 00              .byte13,0
710:  c0b8 4d 45 53  msg8     .asc "message    8"
720:  c0c4 0d 00              .byte13,0
730:  c0c6 93 0d 00  msg9     .byte147,13,0 ;clear screen
740:  c0c9 42 55 46  msg10    .asc "buffer full"
750:  c0d4 0d 00              .byte13,0
760:  c0d6 49 c0 4b  msgadd   .wordmsg0,msg1,msg2,msg3,msg4,msg5
770:  c0e2 9a c0 a8           .wordmsg6,msg7,msg8,msg9,msg10
                      ;
                      ; routine to get file name
800:  c0ec a2 00     fnget    ldx  #0
810:  c0ee 8e 19 c0           stx  fnlen             ; f name length=0
870:  c0f1 20 e4 ff  fnget1   jsr  getin             ;get chars for
880:  c0f4 a8                 tay
890:  c0f5 f0 fa              beq  fnget1            ;filename
900:  c0f7 c9 0d              cmp  #13
910:  c0f9 f0 36              beq  fnget2            ; return
920:  c0fb c9 14              cmp  #20      ; check for delete key
```

```
930:   c0fd d0 10              bne  fnget1x
940:   c0ff ce 19 c0           dec  fnlen
950:   c102 30 06              bmi  fnget1y ; nothing to delete
960:   c104 20 d2 ff           jsr  chrout  ;  delete last character
970:   c107 4c f1 c0           jmp  fnget1  ;  go get next
980:   c10a ee 19 c0 fnget1y   inc  fnlen
990:   c10d f0 e2              beq  fnget1  ; nothing to delete
1000:  c10f c9 20    fnget1x   cmp  #32     ; allow any character for
                                              file
1010:  c111 30 de              bmi  fnget1  ;  name between 32 and 90
1020:  c113 c9 5b              cmp  #91     ;   (ie blank and z)
1030:  c115 10 da              bpl  fnget1
1040:  c117 ae 19 c0           ldx  fnlen
1050:  c11a e0 0f              cpx  #15
1060:  c11c d0 08              bne  fngets            ; no error
1070:  c11e a9 02              lda  #2                ;too many chars.
1080:  c120 20 24 c0           jsr  msgout            ; error msg
1090:  c123 4c ec c0           jmp  fnget             ; try again
1100:  c126 9d 06 c0 fngets    sta  fname,x      ; save character
1110:  c129 20 d2 ff           jsr  chrout
1120:  c12c ee 19 c0           inc  fnlen
1130:  c12f d0 c0              bne  fnget1            ; get next
1150:  c131 20 d2 ff fnget2    jsr  chrout  ; file name entered now
1151:  c134 ae 19 c0           ldx  fnlen   ; so, pad file name with
1152:  c137 a9 a0              lda  #$a0    ;  shifted blanks
1153:  c139 e0 0f    fngetp    cpx  #15
1154:  c13b f0 06              beq  fngetr  ; all done, so return
1155:  c13d 9d 06 c0           sta  fname,x
1156:  c140 e8                 inx
1157:  c141 d0 f6              bne  fngetp
1160:  c143 60       fngetr    rts
                    ;
1180:  c144 a2 06    fopen     ldx  #<fname  ; open4,8,4,"..fname.."
1190:  c146 a0 c0              ldy  #>fname
1200:  c148 a9 13              lda  #19
1210:  c14a 20 bd ff           jsr  setnam
1220:  c14d a9 04              lda  #4
1230:  c14f a2 08              ldx  #8
1240:  c151 a8                 tay
1250:  c152 20 ba ff           jsr  setlfs
1260:  c155 20 c0 ff           jsr  open
1280:  c158 60                 rts
                    ;
                    ;     open disk error channel
1301:  c159 a9 00    open15    lda  #0
1310:  c15b 20 bd ff           jsr  setnam
1320:  c15e a9 0f              lda  #15
1330:  c160 a2 08              ldx  #8
1340:  c162 a8                 tay
1350:  c163 20 ba ff           jsr  setlfs
1360:  c166 20 c0 ff           jsr  open     ;   open15,8,15
1370:  c169 60                 rts
                    ;
                    ;   initialize buffer pointers
1390:  c16a a6 2d    buffinit  ldx  vartab
```

```
1400:  c16c 86 fb            stx  nextfree
1400:  c16e 86 fd            stx  nextshow
1410:  c170 a6 2e            ldx  vartab+1 ;     set free and show
1420:  c172 e8               inx
1420:  c173 e8               inx
1420:  c174 e8               inx
1420:  c175 e8               inx           ;     pointers =
1430:  c176 86 fc            stx  nextfree+1 ;  vartab+1024
1440:  c178 86 fe            stx  nextshow+1
1450:  c17a 60               rts
                        ;   read disk file into memory buffer
1470:  c17b a9 52    getfil  lda  #"r"    ; indicate intent to open
                                            for read
1480:  c17d 8d 18 c0         sta  fnwr
1490:  c180 a9 05            lda  #5
1500:  c182 20 24 c0         jsr  msgout          ;ask for file name
1520:  c185 20 ec c0         jsr  fnget           ;get file name
1530:  c188 ad 19 c0         lda  fnlen
1540:  c18b f0 75            beq  getfnd  ; null file name - so quit
1545:  c18d 20 e7 ff         jsr  clall   ; ensure all files closed
                                            at start
1546:  c190 20 59 c1         jsr  open15  ;  open error channel for
                                            disk
1550:  c193 20 44 c1         jsr  fopen   ;attempt to open disk file
1560:  c196 a2 0f            ldx  #15
1560:  c198 20 c6 ff         jsr  chkin   ; prepare to read error
                                            channel
1570:  c19b 20 e4 ff         jsr  getin
1580:  c19e c9 30            cmp  #"0"
1590:  c1a0 d0 07            bne  diskerr1 ; open attempt failed
1600:  c1a2 20 e4 ff         jsr  getin
1610:  c1a5 c9 30            cmp  #"0"
1620:  c1a7 f0 19            beq  getf2   ; file opened without error
1630:  c1a9 a2 1e    diskerr1 ldx  #30    ; get all of error message
1640:  c1ab 8e 1a c0         stx  charno
1650:  c1ae 20 d2 ff diskerr2 jsr  chrout  ; display error message
1660:  c1b1 20 e4 ff         jsr  getin   ; get next byte from error
                                            channel
1670:  c1b4 ce 1a c0         dec  charno
1680:  c1b7 d0 f5            bne  diskerr2
1690:  c1b9 20 d2 ff         jsr  chrout
1700:  c1bc 20 cc ff         jsr  clrchn  ; clear error channel
1710:  c1bf 4c 7b c1         jmp  getfil  ; go try again
1720:  c1c2 a2 04    getf2   ldx  #4
1730:  c1c4 20 c6 ff         jsr  chkin           ;prepare for i/p
1740:  c1c7 20 6a c1         jsr  buffinit ; set pointers
1750:  c1ca 20 cf ff getf3    jsr  chrin          ;get file character
1760:  c1cd a6 90            ldx  stflag
1770:  c1cf f0 13            beq  getf7           ;no error
1780:  c1d1 8a               txa
1790:  c1d2 29 40            and  #64
1800:  c1d4 d0 29            bne  getf9           ;end of file
1810:  c1d6 20 1b c0         jsr  closf
1820:  c1d9 a9 04            lda  #4
1830:  c1db 20 24 c0         jsr  msgout          ;error, try again
1840:  c1de 20 1b c0         jsr  closf
```

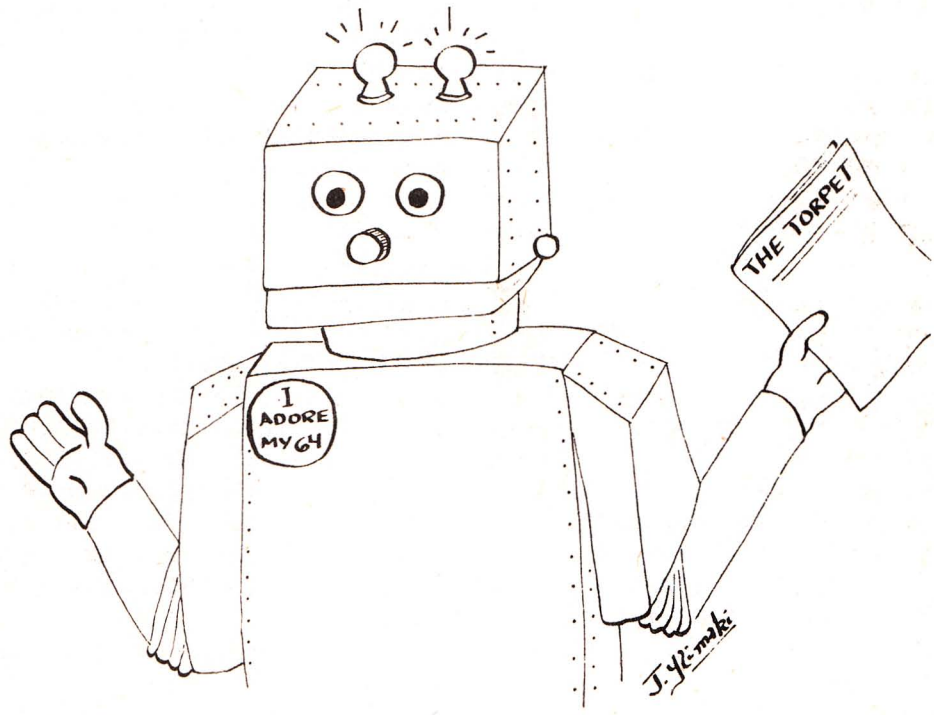
```
1850:  c1e1 4c 7b c1          jmp  getfil
1860:  c1e4 a2 00     getf7   ldx  #0
1870:  c1e6 81 fb             sta  (nextfree,x)        ;save char in
                                                        buffer
1890:  c1e8 e6 fb             inc  nextfree          ;point to next
                                                      character
1900:  c1ea d0 de             bne  getf3
1910:  c1ec e6 fc             inc  nextfree+1
1911:  c1ee a5 fc             lda  nextfree+1
1920:  c1f0 c5 38             cmp  memsiz+1 ; check if buffer full
1921:  c1f2 90 d6             bcc  getf3    ;    no.
1930:  c1f4 a9 0a             lda  #10      ;    yes, so output message
1940:  c1f6 20 24 c0          jsr  msgout
1950:  c1f9 20 1b c0          jsr  closf
1960:  c1fc 4c 7b c1          jmp  getfil            ;go try again
1970:  c1ff 20 1b c0 getf9    jsr  closf
2000:  c202 60       getfnd   rts
                       ;
                       ;
                       ;  put line of text at top of screen
2040:  c203 a9 93    dislin   lda  #147
2140:  c205 20 d2 ff          jsr  chrout  ; clear screen
2150:  c208 a6 fd    disl2    ldx  nextshow ;check for more data
2160:  c20a e4 fb             cpx  nextfree
2170:  c20c d0 0a             bne  disl6   ; go show character
2180:  c20e a6 fe             ldx  nextshow+1
2190:  c210 e4 fc             cpx  nextfree+1
2200:  c212 d0 04             bne  disl6   ; go show character
2250:  c214 a9 07    disl3    lda  #7      ;  all done, so
2260:  c216 d0 2f             bne  dislnd  ; o/p all done msg and
                                             return
2279:  c218 a2 00    disl6    ldx  #$00
2280:  c21a a1 fd             lda  (nextshow,x) ; get char to be shown
2290:  c21c e6 fd             inc  nextshow ; and increment pointer
2300:  c21e d0 02             bne  disl8
2310:  c220 e6 fe             inc  nextshow+1
2320:  c222 a2 00    disl8    ldx  #0      ; ensure that each line
                                             begins
2330:  c224 ec 1a c0          cpx  charno  ;  with a line numer.
2340:  c227 d0 0b             bne  disl9   ;   not first char on line
2350:  c229 c9 30             cmp  #"0"
2360:  c22b 30 db             bmi  disl2   ;      ignore non-numeric
2370:  c22d c9 3a             cmp  #58     ;        at start of line.
2380:  c22f 10 d7             bpl  disl2
2390:  c231 ee 1a c0          inc  charno  ; ok - number found
2400:  c234 20 d2 ff disl9    jsr  chrout  ; put char on screen
2410:  c237 c9 0d             cmp  #13       ; until end of line,
2420:  c239 d0 cd             bne  disl2   ;get next text character.
2430:  c23b a2 00             ldx  #0
2440:  c23d ec 1a c0          cpx  charno  ; check for null line
2450:  c240 f0 c6             beq  disl2   ;    ignore null line
2460:  c242 8e 1a c0          stx  charno  ; reset character count
2470:  c245 a9 06             lda  #6      ; put  sys.... on screen
2480:  c247 20 24 c0 dislnd   jsr  msgout
2490:  c24a 60                rts
                       ;
```

```
2510:  c24b a9 13     setkbb   lda  #19      ; simulate 2 returns.
2520:  c24d 8d 77 02           sta  kbbuff   ; put clear, cr, cr into
2530:  c250 a9 0d              lda  #13      ;   keyboard buffer
2540:  c252 8d 78 02           sta  kbbuff+1
2550:  c255 8d 79 02           sta  kbbuff+2
2560:  c258 a9 03              lda  #3
2570:  c25a 85 c6              sta  kbcnt    ;indicate 3 chars in buffer
2580:  c25c 60                 rts

2600:  c25d 20 7b c1 start     jsr  getfil   ; go get text file into
                                               buffer
2640:  c260 20 03 c2 start2    jsr  dislin   ; put one line of text at
                                               screen top
2650:  c263 20 4b c2           jsr  setkbb   ; simulate return key.
2660:  c266 4c 80 a4           jmp  rbasic   ; graceful return to basic
```

ON THE NEXT PAGE IS A LISTING DATA STATEMENTS TO ALLOW YOU TO USE THE A BOVE PROGRAM IN A BASIC PROGRAM

This takes the place of having to use an assembler and type in the previous machine language listing



COMPUTER DATING IS NOTHING NEW...

I'VE BEEN DATING A COMPUTER FOR YEARS.

LISTING OF DATA STATEMENTS

```
10 print"[clr]sys 49152 to run ... please wait ..."
20 for i=49152 to 49768:read a:poke i,a:next
30 new
1000 data 76,93,194,76,96,194,84,69,83,84,32,76,73,83,84,160
1001 data 160,160,160,160,160,44,83,44,82,9,0,32,204,255,169,4
1002 data 32,195,255,96,10,170,189,214,192,141,55,192,189,215,192,141
1003 data 56,192,162,3,134,154,173,183,192,240,13,32,210,255,238,55
1004 data 192,208,243,238,56,192,208,238,96,13,0,32,13,0,78,65
1005 data 77,69,32,84,79,79,32,76,79,78,71,13,0,79,80,69
1006 data 78,32,69,82,82,79,82,13,0,73,45,79,32,32,32,69
1007 data 82,82,79,82,13,0,70,73,76,69,32,78,65,77,69,32
1008 data 70,79,82,32,84,69,88,84,32,84,79,32,66,69,32,67
1009 data 79,78,86,69,82,84,69,68,13,0,83,89,83,52,57,49
1010 data 53,53,32,32,32,32,13,0,13,82,69,77,32,32,65,76
1011 data 76,32,68,79,78,69,13,0,77,69,83,83,65,71,69,32
1012 data 32,32,32,56,13,0,147,13,0,66,85,70,70,69,82,32
1013 data 70,85,76,76,13,0,73,192,75,192,78,192,93,192,105,192
1014 data 118,192,154,192,168,192,184,192,198,192,201,192,162,0,142,25
1015 data 192,32,228,255,168,240,250,201,13,240,54,201,20,208,16,206
1016 data 25,192,48,6,32,210,255,76,241,192,238,25,192,240,226,201
1017 data 32,48,222,201,91,16,218,174,25,192,224,15,208,8,169,2
1018 data 32,36,192,76,236,192,157,6,192,32,210,255,238,25,192,208
1019 data 192,32,210,255,174,25,192,169,160,224,15,240,6,157,6,192
1020 data 232,208,246,96,162,6,160,192,169,19,32,189,255,169,4,162
1021 data 8,168,32,186,255,32,192,255,96,169,0,32,189,255,169,15
1022 data 162,8,168,32,186,255,32,192,255,96,166,45,134,251,134,253
1023 data 166,46,232,232,232,232,134,252,134,254,96,169,82,141,24,192
1024 data 169,5,32,36,192,32,236,192,173,25,192,240,117,32,231,255
1025 data 32,89,193,32,68,193,162,15,32,198,255,32,228,255,201,48
1026 data 208,7,32,228,255,201,48,240,25,162,30,142,26,192,32,210
1027 data 255,32,228,255,206,26,192,208,245,32,210,255,32,204,255,76
1028 data 123,193,162,4,32,198,255,32,106,193,32,207,255,166,144,240
1029 data 19,138,41,64,208,41,32,27,192,169,4,32,36,192,32,27
1030 data 192,76,123,193,162,0,129,251,230,251,208,222,230,252,165,252
1031 data 197,56,144,214,169,10,32,36,192,32,27,192,76,123,193,32
1032 data 27,192,96,169,147,32,210,255,166,253,228,251,208,10,166,254
1033 data 228,252,208,4,169,7,208,47,162,0,161,253,230,253,208,2
1034 data 230,254,162,0,236,26,192,208,11,201,48,48,219,201,58,16
1035 data 215,238,26,192,32,210,255,201,13,208,205,162,0,236,26,192
1036 data 240,198,142,26,192,169,6,32,36,192,96,169,19,141,119,2
1037 data 169,13,141,120,2,141,121,2,169,3,133,198,96,32,123,193
1038 data 32,3,194,32,75,194,76,128,164
ready.
```

IF YOU OWN A

COMMODORE 64 or VIC-20

HAVE FUN

and

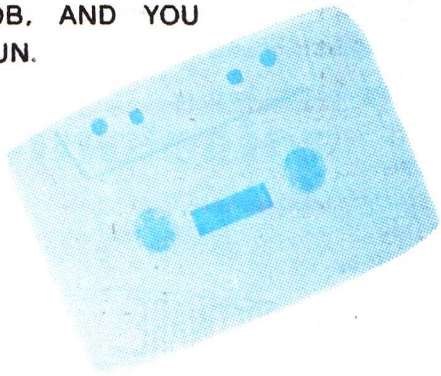
MAKE MONEY

with a

TORPETWARE PARTY

(A New Software & Hardware Marketing Concept
from Torpet Magazine)

DOES SOMEONE IN YOUR FAMILY KNOW A LITTLE BIT MORE ABOUT COMPUTERS THAN JUST GAMES? SAY THEY DO A BIT OF BASIC PROGRAMMING, OR THEY USE A WORD PROCESSOR? HERE IS A CHANCE TO EXTEND YOUR KNOWLEDGE ABOUT COMPUTERS, AND HAVE FUN AND MAKE MONEY AT THE SAME TIME. WE WILL SEND YOU NEW COMMERCIAL SOFTWARE TO TRY OUT AND DEMONSTRATE IN YOUR HOME TO OTHER COMMODORE OWNERS. THIS PLAN COULD MAKE YOU AS MUCH MONEY AS A FULL-TIME JOB, AND YOU WILL PROBABLY FIND IT A LOT MORE FUN.



For Full Details
Write or Call

THE TORPET
Horning's Mills, Ont.
Canada. L0N1S0
or call (519)925-5376

ENHANCE YOUR COMMODORE-64

RAMDISK-64

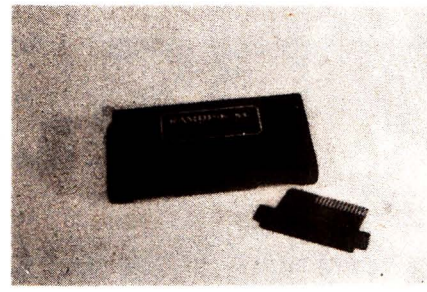
The RAMDISK-64 is a cartridge containing 64K bytes of RAM used to emulate a disk drive. No more long waits for program saves and loads. Use Ramdisk-64 as a second disk drive. Or use the 64K bytes as extra memory for large and data intensive software.

FEATURES:

- 64 kbytes of dynamic RAM
- Includes software to emulate a disk drive
- Loads an 8 kbyte program in less than 1 second.
- Does not use COMMODORE-64 RAM space
- System reset does not erase files in RAMDISK-64.
- Small size only 3"x5.5"x0.5"
- Useable with other cartridges

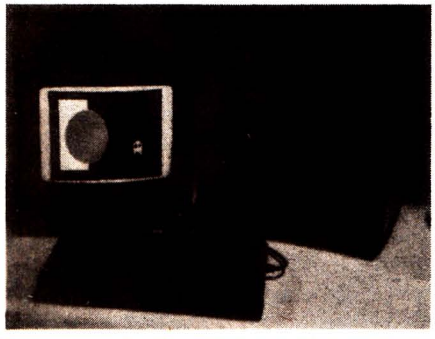
APPLICATIONS:

- Time saver where disk usage is high:
  - Software development
  - Large programs using overlays
- Software requiring large data storage. Use with data bases or spreadsheets. Save multiple graphic screens.
- Use as a second drive. Minimize diskette swapping.



MORE ABOUT DISK EMULATION SOFTWARE

- Allows 16 directory entries or 63.5 kbytes of storage.
- Compatible with BASIC commands OPEN, CLOSE, GET#, INPUT#, PRINT#
- Supports PRG and SEQ files
- Treat RAMDISK-64 as device 15, user changeable.
- OTHER FEATURES TOO NUMEROUS TO MENTION HERE.



VIDEO-80

The VIDEO-80 is a high-quality 80-column cartridge.

FEATURES:

- 80 columns x 25 lines display
- 256 character set with full Ascii and reverse letters
- Flicker-free crisp display even during scrolling
- Compatible with BASIC and the KERNAL
- Includes customizing video routines like scrolling, insert and delete line, address cursor, and screen blank
- DUAL SCREEN: Connect the normal C-64 video output to a color monitor AND connect the VIDEO-80 video output to a B/W or green screen monitor. Text will appear on the B/W or green screen monitor and the color graphics on the color monitor. A NECESSITY FOR GAME DEVELOPERS AND GRAPHICS PROGRAMMERS.
- Compact size only 3"x5.5"x0.5".

APPLICATIONS:

- Word Processing
- Spreadsheets
- Terminal Emulation
- Software Programming. Eliminate irritating line wrap around.
- Graphics development. Use VIDEO-80 for text and C-64 video output for color graphics simultaneously.

The RAMDISK-64 and VIDEO-80 use the COMMODORE-64 expansion slot. An optional extender board ($10) or motherboard is required.

PRICES

RAMDISK-64 ..... $149.
VIDEO-80 ..... $99.
Extender Board ..... $10.

Calif. residents add 6% sales tax. Add 5% shipping and handling. VISA, MasterCharge, COD accepted. Personal checks require two weeks to clear. Order by phone or mail.

PRICES QUOTED IN U.S. FUNDS

P Technologies

1555 Riverpark Drive, Suite 206
Sacramento, CA 95815
(916) 920-3226
8:00 am to 5:00 pm PST, Mon-Fri.

EXCLUSIVE CANADIAN DISTRIBUTORS - B & R ENTERPRISES & ASSOCIATES
29 IRVING DR. - PEFFERLAW, ONT. - L0E 1N0 - CANADA - (705) 437-3187 EX. 21
CANADIAN DEALERS INQUIRIES INVITED.

THE BEST OF THE TORPET available soon

Mail orders to: The TORPET
Horning's Mills, ON.
LON 1J0

DISK MYTHS YOU SHOULD KNOW ABOUT!

by Tom Van Flandern
Washington, D.C.

Reprint from WINePEG V.2 No. 4

WINePEG is a newsletter of the Windsor and Area Pet Educators Group.

Is the price of blank diskettes a constant drain on your budget? Read on! You'll be glad you did.

Discussions of the subject of diskettes usually result in the propagation of "disk myths", or statements about diskettes which have three attributes: 1/ They originate from diskette manufacturers or dealers, not users; 2/ they are all reasons why you should pay more for your diskettes; and 3/ they are untrue. Lets consider the most common of these myths.

One-Sided Myth

You have probably been exposed to the controversy over using one or both sides of your "single-sided" diskettes. I have often heard the myth repeated that the manufacturers put their label on whichever side of a diskette that their surface quality tests. By implication, the other side may have failed such a test and therefore may be expected to be of inferior quality. Sounds plausible, doesn't it? Cuts manufacturing costs, and why certify both sides when only one is usable as the disk is sold? There is just one problem with the theory - the box of diskettes doesn't know what type of computer or drive it is headed for use on. Did you know that Apple disk drives always write on the bottom side of your diskette? Commodore disk drives do the same; however, there is no standard among computers. Some have single-sided disk drives which write on one side; others may write on the other side. Manufacturers are therefore obliged to certify both sides of diskettes with equal care.

Two-Sided Myth

What keeps you from turning your diskettes over and using the magnetic surface on the other side? There is a small rectangular notch along one edge, centered at 1 5/16 inches from the top edge of your 5 1/4 inch floppy diskettes. This notch permits your disk drive to sense that it is okay to write on the disk. If you cover this notch, the disk is write-protected. To make the other side usable, just punch a similar hole along the opposite edge at the same distance from the top. Turn the diskette over, insert it into the drive and use in the normal way. The shape of the notch is not important - circular or rectangular are equally good - but it must be at the correct location, about 1/4 inch wide and not quite as deep. Use an ordinary hole punch for good results. To get the location correct, just turn over another diskette and line it up with the one to be punched. For mass production, make a mask 5 1/4 inches long which can be placed quickly over the disk to show you where to punch. Don't be concerned if you get the hole slightly too large. Your chances of damaging the diskette are small with ordinary care and are less from making too large a hole than they are from using too crude a cutting instrument, causing the diskette to be pinched inside its cardboard jacket.

The disk myth in this connection is that you risk losing data on the original side of the disk if you write on the other side in a one-sided drive. The reason cited is that magnetic particles will accumulate on the pressure pad which presses against the side opposite the read/write head, and these can destroy information on the side they come in contact with. The principal argument against this theory is empirical - it just doesn't happen, at least not over a period as short as a few years in ordinary usage. (See caution below under "The Cleaning Kit Myth" however.) The failure rate for diskettes used one-sided and two-

sided is statistically indistinguishable, resulting in an interesting correlation. The probability of a micro-computer owner using his diskettes two-sided is directly proportional to his experience. Almost all users eventually try this, and the best proof of its effectiveness is that they stay with it. The most experienced owners, with the largest files, almost all use their diskettes two-sided, and smile knowingly at the novices who are reticent because "if it were that simple, the manufacturers would tell you so"!!

Double-Density Myth

This disk myth is insiduous, because the manufacturers allow the consumers to fool themselves and simply fail to provide them with information needed to correct the myth. Double-density diskettes cost more because they have a thicker magnetic coating. So they must surely be better, right? Why not keep your really important files on double-density diskettes? Woe to you, naive and trusting user. The purpose of the double-density diskettes is to support disk drives capable of generating a stronger magnetic signal than normal drives. This is usually needed if more bits are to be written per inch, but is quite unnecessary for the information density at which normal disk drives operate. More importantly, though, since the signal generated by normal drives is not strong enough for double-density diskettes, you actually have a slightly higher risk of losing those valuable files if you wrote them on a double-density diskette!!

Hub Ring Myth

Some diskettes come with hub rings, and this too is supposed to be worth paying extra for. Hub rings are circular bands on the inner edge of your diskette which provide extra strength to that edge. Their main function is to keep the inner edge from getting crunched if the diskette is off-center when the lid and pressure pad are lowered after the diskette is inserted into the drive. With just a minimum of care however, the lid can be closed slowly and lifted and closed again if it meets resistance, so as not to damage the diskette. Another recommended practice is to boot your disk and start it spinning before lowering the door lid. (As far as I can tell, this is NOT possible for those using the 1541 single disk drive. JM) This not only aids self-centering, but also prevents the read/write head from pressing against the disk surface as it retracts for recalibration (the clacketty noise you hear). In other words ordinary good disk-handling practice (which even children can be expected to follow) will allow the diskettes a chance to self-center and prevent damage. The problem caused by the hub rings is that, if the diskette has any tendency to bind in its jacket, preventing it from gaining full rotation speed, it is easier for it to slip with the hub rings than without. If you ever try a disk-speed test and occasionally see some measures go off the scale, this is usually from binding up and may be exacerbated by hub rings.

The Name Brand Myth

Occasionally a brand of diskettes in its entirety or a particular batch of diskettes from some well known manufacturer, will be flawed and produce much user grief. However, there is a lot of incentive for manufacturers who want to stay in business to prevent this from happening and most are successful. Once the diskettes pass the surface certification tests, if they are properly shipped and handled, they are essentially equally good, regardless of name brand or claims to the contrary. Almost all diskette failure is due to handling problems (see below). Failure rates of factory-shipped diskettes are about 12 per 1000, on average, with little variation between brands and no correlation with price. The myth here is that paying more for a name brand buys a tangible benefit. In fact, many generic brand diskettes are available, often made by the same big-name manufacturers but without the name brand label, for much less cost than the identical diskette with the label pasted on it. Is the label really worth that much extra cost to you? There is also the question of whether a manufacturer will stand behind its guarantee. Apple Avocation Alliance recently reported that Verbatim refused to honor its diskette guarantee and criticized the Apple organization for selling Verbatim's "too cheaply".

The Cleaning Kit Myth

"Buy a cleaning kit for your disk drive. Clean the read/write heads at least once a week." Before I knew any better I bought just such a head cleaning kit. At the time I wondered at the important notice on the box, which I quote in part: "Neither seller nor manufacturer shall be liable for any injury, loss or damage arising out of the use of the product. Before using, user shall determine the suitability of the product for his intended use, and user assumes a risk and liability whatsoever in connection." I assumed, as most people must, that this was just legal mumbo-jumbo to protect the manufacturer from frivolous lawsuits by incompetent users. After all, the product was being widely sold for the purpose of cleaning disk drive heads, and that was surely a desirable end. Wrong again! I began to have one diskette failure after another and it was several months before I realized the correlation with use of the head cleaner.

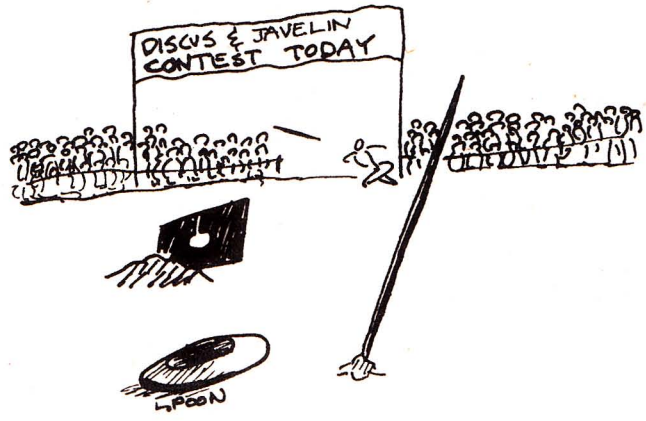
The sad truth is that the cleaning fluid used with the kit is a strong solvent. The recommended method of application, results in the pressure pad getting soaked with solvent. If you then use a diskette in the drive, the magnetic surface on the other side of the diskette is scoured by the solvent and actually dissolved in the process! The damage can be so extensive that it may no longer be possible to initialize the damaged surface. Of course if I had not been using my diskettes two-sided, I might not have discovered the problem. But I now know that, in truth, head cleaning usually needs to be done at intervals of one to four years, not weeks, and is easily done with alcohol applied directly to the head, without damage to either pressure pad or diskettes.

Proper Care and Handling

Most diskette failure is caused by improper care and handling, rather than anything under the control of the manufacturer. Of course, diskettes must be kept away from magnetic fields, such as emitted by some TV's and certain other electronic devices. They must be kept clean and dry. And the importance of never writing on a diskette label with an object which can apply pressure to the magnetic surface below cannot be over-emphasized.

Perhaps the single most common cause of random diskette failure not caused by disk drives is binding in the cardboard jackets. This is why you are advised to store diskettes vertically and avoid the temptation to stack them horizontally. Anything which applies pressure to the jackets (including crowded storage of diskettes, horizontally or vertically) can cause binding, which prevents the diskette from spinning at full speed continuously while in use, which causes intermittent failures.

I hope the preceding information proves useful to you and saves you money as well as headaches with your diskettes. *



A TINY PRINT HINT

Elizabeth Deal
Malvern Pa

Sometimes a screen design requires using all forty positions of a line. You can define a 40-character string and print it with a semicolon at the end, falling through to the next line. Don't do it!

There is a pitfall in this method, causing difficulties in debugging. You cannot reliably overprint such a screen. The computer keeps track of the double lines and in so doing messes up your best intentions. You issue one cursor down, for instance, and try to print your name ... and it invariably (or one out of two times) ends up precisely where you didn't want it.

The reason is that the computer is doing its best to keep its house in order. It follows the pattern of single or double lines established by the first PRINT, and the subsequent overprinting follows the line length already known. The table of line lengths is kept in page zero. See memory map under "screen line link table" or a similar name, if you're interested in details.

There is a way to print forty character strings by doing a little trick with the insert-character. To understand how this works, try this: in direct mode, print 39 stars. Now move one cursor left and push the INST-key (that's shifted-delete). Type a star in the gap remaining, and you have a clean, 40-character line.

As an illustration of using this sort of thing for more than one line on the screen, use the framemaker routine. The same process is used as the previous, direct mode, procedure. Note that each line ultimately ends in carriage return (no semicolon). This keeps us out of trouble. Following the last line of the program you can insert some code of your own to print the screen in any position you like, or you can use what's there already. If you plan to print lines of 40 characters, again, use the insert procedure. For this demo I have not done so.

This method is particularly handy on the C64. On the PET, we often print 39 characters, and poke the fortieth one onto the edge. Often however, there is no need to bother.

On the C64 this is more critical especially when the edge and background colors are different. The screen may look sloppy if the fortieth position is empty. Furthermore, the C64 opens up lines on the screen when the cursor is in the last position, useful in program editing, a headache in neat screen design! Since poking both the screen and the color memory isn't fun, the framemaker method can be a viable alternative if you need it.

```
10 REM FRAMEMAKER....ELIZABETH DEAL
20 F$=CHR$( 157)+CHR$( 148)
30 TB$="****************************************":REM 39* IN ""
35 TB$=TB$+CHR$( 145)+CHR$( 13)+CHR$( 148)+"*"
40 MD$="*                                      *"+F$:REM *,37 SPACES,* IN ""
50 PRINTTB$:REM TOP LINE
60 FOR J=1 TO 23:PRINTMD$:NEXT J:REM MIDDLE
70 PRINTTB$;"[3]";:REM BOTTOM
80 REM HOME IS USEFUL FROM BOTTOM LINE
90 PRINT"[5 DOWN]LINES":REM 5 DOWN IN ""
```

CENTIPEDE

by ALYSSA KRAUSS (13 yrs.)
and GEOFF KRAUSS
LATHAM N.Y.

The arcade game "Centipede" recently became available in cartridge form for the C64. Like the arcade game, it has all the characters of the Enchanted Mushroom Patch, including the Dreaded Bug Blaster that destroys bugs and pieces of mushrooms and the Centipede that zig-zags back and forth each time it bumps into a mushroom and splits each time it is hit by a Bug Blaster blast. The Frenzied Fleas create more mushrooms as they fall, and can gobble up your Blaster whenever they land. The Jumping Spider, which isn't blocked by mushrooms, can also pounce on your Blaster and destroy it. Finally, the very dangerous and poisonous Scorpion (which I have never seen on the arcade version) can dash across your mushroom patch and poison any mushroom just by touching it! When any Centipede touches a poisoned mushroom, that Centipede goes insane and heads directly for your Blaster.

Compared to the version of this game in the video arcades, I like this one a lot better. It has better graphics, including numerous sprites (both single and multi-colored), and is more challenging. The mushrooms are not destroyed by a single blast, so it is harder to clear an area than in the arcade version. This adds to the difficulty and the challenge. The sound capabilities of the C64 also make a big difference. The Centipede, the Spider, the Fleas and the Scorpion all have their own "songs", and it is very interesting to hear several different melodies being played at the same time. After a few rounds of play, you learn to recognize which menaces are on-screen by the sound combinations, and can change your strategy; this is very important, as the number of objects, and especially moving objects, on the screen is large and is another tribute to the great graphics ability of this game on the C64. The only thing I don't like about the game is that you finally have to turn it off.

On a scale from one to 10, I rate this game at 9-1/2. I think if anyone wants to buy a good arcade game for their C64, this is one of the best you can get. This game cartridge is in the $40.00 (U.S.) price class (from Atari-Soft Inc.).

Dad adds: Unlike the arcade version, C64 Centipede allows forward/backward movement of The Blaster, as well as left/right movement; this allows a greater latitude in attempting to escape Menaces. However, the game is set up for a joystick in port 2 (nearest the power switch), while most other C64 games are set up for the joystick in port 1; this port change is fine if you have two joysticks, but is somewhat of a nuisance if a single joystick has to be changed between ports (it also increases the chance of eventual port/connection failure). Also, unlike most other C64 games, a replay <u>cannot</u> be activated by the "fire" button; each play requires that at least two function keys be used (either F1 for one player or F3 for two players; followed by F1 to start on-screen action). The switch between players is sufficiently fast enough to make a two-player game much more difficult than a single-player game. Centipede addicts, such as I, might well have preferred a single-player fire-button-repeat "normal" game mode, with a two-player mode selectable by keyboard intervention. Since a clear view of the VDT is really essential for this game, it is somewhat disconcerting to have to be close enough to the C64 to hit F1-F1 to start each game, and then shift visual focus back to the VDT to play. My personal rating is 8.5/10 (the 9.5 rating above is the "family" average of the 10/10 rating by Alyssa and her eight-year-old brother, and my own rating!).

CLASSIFIED ADS

5 Cents per word, with $1.00 Minimum.
Payment in advance to:
TORPET CLASSIFIED
Horning's Mills, Ont.
Canada L0N 1J0

The TORPET reserves the right to limit ads from businesses placed in the classified column.

FOR SALE

Heavyweight Championship Fight! Strategy (not arcade) game on tape for Commodore 64. Manage the sixteen greatest boxers of all time. **NCAA Basketball**, due in January. **Relative Baseball**, due in April. $11.95 per game, $2 postage/handling per order. No Limits Software, Box 2605, Madison, Wisconsin, 53701. s

---

Nelma Persona: With CP/M, Wordstar, Calcstar and Modem 7. Used approx. one month for software evaluation. $2300 + P.S. cart, acceleration of a small car and other experiments. All on disk ready to use. $39.99 each. Komputer Science Kits, Rednersville Road, R.R. 1, Carrying Place, Ontario, K0K 1L0.

Re-Ink your used printer ribbons for only pennies, again and again, with the Mac Mark II motorized re-inking machine. Fully guaranteed, for any model, includes ink for 20 inkings. To order, send cheque or money order for $84.95 + P.S.T., where applicable, to ReInc., Box 5555, Concord, ON., L4K 1B6, or phone (416) 635-6820/928-0691. Visa and Mastercard accepted. Please allow 4-6 weeks for delivery. Specify make and model of printer. $ave Bundle$. jul

---

Stuntman Spelling A quality field-tested educational game for kids of all ages. Purchase 1 program and make up to 25 free copies for school or home. Our policy on all products. Stuntman for VIC, or PET $24.95 to: Synchronizing Education and Games, 668 Sherene Terrace, London, ON. Canada N6H 3K1. au

---

Uncrasher Switch for PETS: Allows you to regain the cursor without shutting off the computer and losing your program. A very simple installation done in minutes. $9.99 each.

Phototimer Kit for PET or C64: Utilizes the computer to time Physics experiments, e.g., period of pendulum, velocity of cart, acceleration of a small car and other experiments. All on disk ready to use. $39.99 each. Komputer Science Kits, Rednersville Road, R.R. 1, Carrying Place, Ontario, K0K 1L0.

8032, like new: $595.00 U.S. plus shipping. E.W. Pedersen, 3579 S. Marine View Drive, Greenbank, WA 98253, U.S.A. Phone (206)678-4789.

PET 4032 (Fat Forty) with graphics keyboard. Dataset and some games. $400 or best offer. Dennis Hogan, Centerville, UT. 801-295-8577.

Diskettes, 5-1/4: SSSD $28.50; SSDD $29.50; DSDD $39.50. Box of 10. Lifetime guarantee.

5-1/4 or 8" Head Cleaning Kits -- $35.00. Also complete line of software for the Commodore 64. Call 416-871-4155. Visa-Mastercard. Add $2.00 shipping and handling. Quantity discounts. Send cheque or money order to Pedlar Discount, 84 Jarvis Street, Fort Erie, Ontario. L2A 2S4.

Computer Desk: Compact functional design, will organize your total computer system and provide a large work area on one level. A shelved area provides plenty of storage space for software and manuals. Complete unit cut from one sheet of plywood, detailed plans and instructions. Send $5.00 to J. Barnard, P.O. Box 191-D, Mount Forest, Ontario. jun

---

Video Poker Fans: Author of "Poker", March Computer's Gazette, has new version for unexpanded VIC-20. Same great features as the original. Greatly-improved C64 version also available. Send $6.50 for tape or $8.00 for disk (U.S. funds), to: A.J. Kwitowski, 302 Euclid Ave., Glassport, PA 15045.

Jinsam 1.0 Data Manager software package for 32K PET and 2040 disk drive. Like new and includes original ring binder with documentation, ROM chip and all disks. Never used. U.S. $100.00. Lyman Duggan, 813-885-3996.

VIC-20 Game Players: Due to tremendous response and customer satisfaction, I am again selling three other games that I have created, each more challenging than the next. **Vic Lawn:** Mow the lawn as fast as you can but, after each set, you soon realize that your mower is guzzling more and more gas. Can you clear the 13th set? I've never done it! **Super Breakout:** If you enjoy fast action, this is the game for you. As you destroy each brick, your game becomes faster and faster, making skill a necessity. **Egg-Bert:** In this game, snakes are hot after your trail as you move Egg-Bert, filling the diamond with colored blocks. (Excellent color and sound.) All games on cassette and run on standard VIC. Please send cheque for $13.00 (for all three) to Mark Idzik, R.R. #4, Kenilworth, Ontario N0G 2E0, Canada. Guaranteed fast reply. P.S. If my other three games are also desired -- Junior King, Cave Patrol and Viper Pilot -- at the same low price, please refer to game details in January edition of TORPET.

Commodore SuperPET computer with 4040 dual disk drive (2020 upgraded), manuals, cables, Black Box ABC switch, and MSD IEEE C64 interface. This hardware can be used to inter-connect two computers. $950.00 U.S. Also BPI Accounting System (includes ROM chip in SuperPET and manual). $150.00 U.S. Call (919)424-1670. jun

BBS Software! Runs on Commodore 64 with disk drive and auto-answer modem. Printer is optional. BBS program, SYSOP utilities and complete documentation, all for $95. Call 914-623-4248 to try it out! Send cheque or money order to: Michael Kleinert, 9 Crabapple Lane, Nanuet, NY 10954.

Commodore Computer Owners... Still paying those high prices for your computer programs? Well, Micro-Wood has famous brand-name games, utilities, business and educational programs at super low discount prices. Come to Micro-Wood for our everyday low discount rates and get more for your money. Send today for free details: Micro-Wood Software, Dept.-T, Box 243, Morton, PA. 19070, U.S.A. jun

---

C64 and VIC-20 Programs for sale: Top-selling games such as Beach Head and Jumpman, as well as hundreds of others. For a complete listing, send $3.00 (deducted from your first order), to Todd Chamberlain, care of Kilroy Software, 306-598 28th Street East, Prince Albert, Sask., Canada. Phone (306)764-0158. jun

---

C64 Word Processor. Quickwriter II, $29.99 (U.S. funds, includes post, handling). 100% machine code. Features: works with tape/disk, any printer/interface. Easiest word processor to use. Easy access to printer's special features. Pre-view displays text as it would be printed out before printing (save paper). Form letters, copy/transfer/delete blocks of text, memorize/append blocks, auto page numbers, right justify, text of infinite length, send any disk command while in the word processor. Available tape or disk.

C64 Machine Utility: Power plus gives over 40 new commands including: screen dump, un-new, DOS commands, machine language monitor, auto line numbers, find, change, renumber, delete, etc. Cost $19.99 U.S. funds, includes post, handling. Available tape or disk. Send cheque to EDUCOMP, 2139 Newcastle Ave., Cardiff, California, 92007. Specify t/d.

Games for PET, VIC and 64 computers on tape, disk and cartridge. Including: Space Intruders, Millipede, Wallbanger, Deadline, Frogger, Supersmash and many more. Also, a trackball for VIC or 64. Send U.S. 20 cent S.A.S.E. to Robert Gruskiewicz, 417 Susquehanna Ave., Wyoming, PA 18644, to receive complete list.

Attention, PET, VIC and C64 Owners: Our new line features reasonably-priced, top-quality programs, including teacher utility (grading, attendance, test-making, others); Statistics series; educational games; drill; and a stock options program. Additional programs available in all subject areas. Specify machine. Accessories catalog (disks, word processors, etc.) available upon request. Contact: Melcher Software-TP, P.O. Box 213, Midland, Michigan 48640. (517) 631-7607. jun

Commodore PET 4016, upgraded to 32K, and 2031 drive with IEEE cable. In excellent condition. Will not sell separately. Asking price: $1,000.00. Contact Guy Leger, 1022 Wilson, Montreal, P.Q. H4A 2T2, or call 514-481-0884 between 7 and 10 p.m.

Super Value for Telecommunications Buffs!!! McTerm RS232 communications package for PET/CBM. ("A very effective package", as stated in TORPET's special issue, December 1983, **McTerm and the Signalman Mark IV** by Dennis Irvine). Includes cable and connectors, ROM chip, 80-page manual and 8250/8050/4040 diskette. List: $290. Sale: $145. Inventory balance, guaranteed original dealer stock. Phone now: (514)465-6056. Addax Micro Inc., 208 Taschereau, Greenfield Park, Quebec, Canada, J4V 2H4.

VIC-20 computer, datacassette, 1525 printer, modem, 3-slot and 4-slot expansion boards. With expandable RAM to 29K Basic and 40K Machine Language. Utility cartridges: Machine Code Monitor, Programmer's Aid, Super Expander. Eight-game cartridges including Sargon II Chess, Gorf, Omega Race plus other software and games on tape plus complete TPUG VIC-20 library. Best books on 6502 machine language and VIC-20 including Intro to Basic course. $750 complete. Call 416-743-5594 after 6 p.m.

SuperPET, 4040 drive, Epson MX80 F/T with Paperclip word processor installed. Versions 1.0 and 1.1 Waterloo BASIC, FORTRAN, APL, PASCAL, DEVELOPMENT, and new COBOL 1.0. All manuals, plus back issues of Transactor with additional reference books. Custom desk also available. Best offer, hardly used, excellent condition. Call Oshawa, 728-6829 after 5, Doug.

Gemini-10/Microworld 302 Owners: Don't buy another interface to get listings of Commodore/TPUG basic 1541 disk programs from your C64! Printer or screen option. Full conversion of cursor controls <CD>, etc. Send $19.95 (U.S.) PPD to Computer Survival Skills, 8373 Bevan, San Gabriel, CA 91775, for Listergem program on disk. jun

---

Modem Owners — It's Here!! A complete directory of BBS (Bulletin Board System) telephone numbers. This guide contains thousands of active BBS numbers you can access from home, as well as pointers on accessing BBS systems. Get programs, information and user help from across the country. A must for any modem user, a telecommunications essential. Ask for the "BBS Reference Guide". Send $9.95 to Datacomp, 2230 "K" Street, Eureka, California, 95501. o

---

PET 8k computer with built-in datasette. Best offer. (416) 884-7458

Introducing Suzuki Omnichord MDL OM-27, portable computerized music synthesizer. State-of-the-Art microtechnology makes mistakes a thing of the past. Now, for musician or non-musician, the future arrives today.--for everyone, the excitement, the satisfaction, the positive reinforcement of being a musician, a performer. Strum it like a guitar even though it has no strings to tune or break. Omnichord's touch-sensitive Sonic-Strings are activated by the touch of your fingers. Or play it like a piano or organ, producing beautiful vibrant tones from its major, minor and seventh chords. With a myriad of possible playing variations from the 27 chord buttons, 6 programmed rhythms, variable tempo control, auto walking bass, manual, memory and sustain keys, and the revolutionary electronic Sonic-Strings (pre-set to coordinate with each chord)-- the educational potential is limitless. All these features, which can be played individually or in exciting combinations, are basic and self-explanatory. Never needs tuning, but a pitch control is provided for playing "ensemble". Completely portable -- weighs only 4 pounds and operates on "C" batteries or the optional A/C adapter. With its built-in speaker, it can be played anywhere. Or, for a concert hall sound, play it through your stereo system, amplifier, or PA system. Manufactured from high-impact unbreakable plastic with one-year factory warranty and instruction manual included. SASE for brochure. LLAT Enterprises, P.O. Box 2517, Covina, CA 91722.

Now! Specially priced for a limited time only! A brand new game for your Commodore 64 frm Bailey/Soft -- **Pyramids of Egypt** (TM) -- an adventure game made especially for your Commodore 64!!! You will travel through the bowels of an ancient pyramid all the way to outer space, complete with sound and color graphics. Disk only. Get yor copy now; we can't keep this low price for long. Send your cheque or money order for $8.95 to: Bailey/Soft, 2600 Williamsburg, Bartlesville, OK 74006. Phone (918)333-6343.

Super VIC-20 Software! Games similar to Pacman, Centipede, Frogger, Berzerk, Missile Command, plus many, many more hit games! We also have Home Utilities and a number of educational programs. Write to us for a free catalog! R-T-M, 3025 Pl. Byzance, Brossard, Quebec J4Z 2T8. If we get your request before July 31, you will receive a free software bonus gift!

Never-used Visicalc for PET/CBM 2001 and 8032 (32K) on DOS 1.0. Complete documentation + ROM chip, $109 or best offer. Call Mr. Ryan, (416) 363-3074.

Commodore 64 Software for re-sale: (all programs in excellent condition) Jumpman (disk)-$49.99, Temple of Apshai (disk)-$49.99, Spritemaster (disk)-$54.99, Blue Max (disk)-$49.99, Tel# (416) 878-4137 (ask for Gary) Call between 9am and 5pm Mon-Fri.

WANTED

Commodore 8032 computer, 2031, 2040, 4040 or 8050 disk drive. Call (416) 366-2130.

Commodore 4040 drive (will consider 2040 or upgrade). Please indicate drive type, condition, age, price and record of any service performed. Will consider as part of system. Paul V. Skipski, 4782 Boston Post Road, Pelham, N.Y. 10803. 914-738-6096.

HELP WANTED

You can become rich as a computer operator. If your program meets our standards, we may publish it with the royalties directly payable to you! If you always wanted to have a career as a computer programmer, program designer or graphics designer, send your work on disk or cassette for the Commodore 64 or VIC-20 to Kilroy Software, Program Review Division, 306-598 28th Street East, Prince Albert, Sask., Canada S6V 1X7, or call (306) 764-0158.

COMPUTER PUNS

Jim Butterfield's cats are now famous for their attraction to his computers and CAThode ray tubes. I also have a feline ON LINE, but mine has taken to sprawling on my printer. Did I mention that she was a TABby...a little spaced out at times?

Farmers may soon be using a FAT FORTY on the back forty as Commodore technology sweeps the nation.
- farmers will be DISKing the land, planting RANDOM SEEDS and raising SERIALS
- fertilizer needs will be handled with an ELECTRONIC SPREADSHEET
- by using data files, any farmer can be out standing his FIELDS!

Don't be alarmed if someone tells you there is a QUICK BROWN FOX running in their computer. It's probably chasing the RABBIT cartridge. More critters are in the computer ads now than on Noah's ark.

Ylimaki

COMAL

TIRED OF BASIC PROBLEMS?

You can get the structure of Pascal and the ease and friendliness of BASIC with COMAL. The language of the future is here today for PET/CBM computers. The COMAL (COMmon Algorithmic Language) microcomputer language is creating interest all over the world. It is well established for educational uses in both Denmark and Ireland. COMAL combines the simplicity and ease of BASIC and the powerful structure of Pascal. COMAL requires no separate editor, yet is line number independent for fast execution. COMAL checks all syntax as the code is entered, so you will never find a syntax error when you run your program.

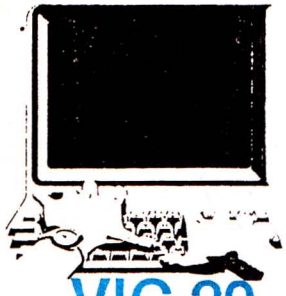
COMAL BOOKS
and
PROGRAM DISKS

Write to:

COMAL INTEREST GROUP

P.O. Box 18432, Las Vegas NV 89114

USER CLUBS



VIC-20 USERS

Get access to hundreds of public domain programs.

Tape $5.00 • Disk $8.00

Informative monthly newsletter, cartridge rentals, contests and more!

3 months trial membership . . $ 3.00
12 months membership $10.00

BUDGET WISE COMPUTER USERS GROUP
17 Chaplin Avenue
St. Catharines, Ontario L2R 2E4

Write for information, enclosing $1.00 to cover postage and handling.

VIC 20 is a trademark of Commodore Electronics Limited.

UNITED STATES

COMMODORE USERS GROUP
P.O. BOX 2310, ROSEBURG, OREGON 97470

Join the largest VIC-20 / COMMODORE 64 users group in the United States.

Individual and chapter support 12 month membership:

U.S. $20.00 - USA & Canada
U.S. $30.00 - Foreign

NO GROUP OFFERS SO MUCH

The PET Educator's Group

JOHN MOORE
c/o P.O. Box 454,
Station A,
Windsor N9A 6L7

ADVERTISER INDEX

THE TORPET TEAM

— Front row, left to right: Jean Beach, Sue Spires;
Back row: Gottfried Walter, Darrin McGugan, Bruce Beach.

Descriptions and multiple sources for over

1000 FREE PROGRAMS

for the

COMMODORE 64* & VIC-20*

CHIPP cartoon strips that explain BASIC for the

BEGINNING PROGRAMMER

Articles for the

INTERMEDIATE PROGRAMMER

on MACHINE LANGUAGE COMPILERS AND OTHER LANGUAGES

Complete MEMORY AND CROSS REFERENCE MAPS printed

IN LARGE TYPE

for the

ADVANCING PROGRAMMER

along with complete instructions for
MICROMON, WEDGE-64 and BYTEDS
The three most powerful public domain utilities.

A thorough compilation on

COMMODORE COMPUTER MUSIC

by the foremost authorities.

User documentation for the

200 BEST GAMES, UTILITIES

and other programs in the public domain.

Articles that give simple and complete explanations of TELECOMMUNICATIONS, DISKS, PRINTERS, TAPE, WORD-PROCESSING, DATABASES, SPREADSHEETS

PLUS MUCH MORE

IN THE BEST OF THE TORPET

Available From:

Copp Clark Pitman Ltd.
495 Wellington Street West
Toronto, Ontario M5V 1E9
(416) 593-9911

"Commodore-ready", . . . and ready for you NOW!

"Cardcorder" DC/1, Data Cassette Recorder/Player

Introducing the "CARDCORDER", Model DC/1, the Computer Cassette that is "Commodore-ready," designed for storage and retrieval of computer data efficiently, economically; with consistent performance. Yet, this fine CARDCO product is priced lower than any similar product with special quality features.

Includes standard connector which is "Commodore-ready"; LED "save" indicator light which confirms data recording on to the tape; handles up to 120 minutes (60 minutes on each side) of any standard tape including existing pre-recorded commercial as well as personal data tapes intended for use with Commodore Personal Computers; ready to go . . . just plug it in and record efficiently.

CARDCO's "CARDCORDER" COMPUTER CASSETTE is a quality data cassette recorder/player in an attractive polystyrene case, with all the standard cassette functions: record . . . play . . . rewind . . . fast forward . . . stop and eject . . . pause. A solid-state designed product of the finest components with auto-stop.

The "CARDCORDER" DC/1 carries a 90 day warranty to original owners.

All CARDCO products are available at your local dealers.

cardco, inc.

313 Mathewson Wichita, Kansas 67214 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."