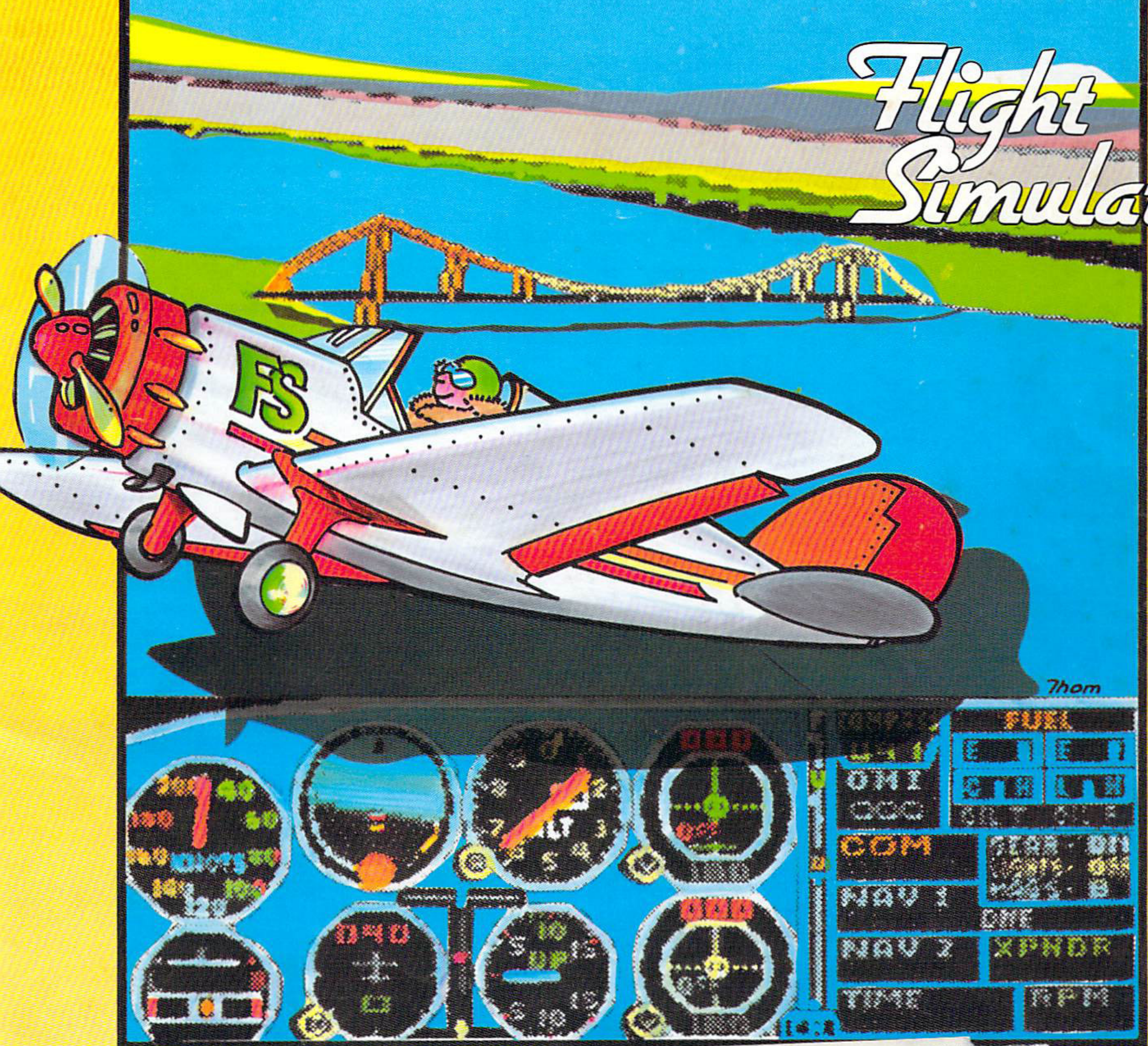
The Magazine For All Commodore Computer Users.

TPUG Magazine

$2.95

Issue No. 25

Flight Simulation



56698 70976 25

BACK ISSUES

DON'T MISS. ORDER NOW

TORPET 26
Feature: Complete educational library listing explained. Also: Commodore silver anniversary; Programming Function Keys; C-64 Logo.

TPUG FEBRUARY 1984
Feature: Canadian bulletin board numbers. Also: Computers and copyright; Disk inner secrets (Jim Butterfield); Computer Colours; (C) T5 documentation.

TPUG MARCH/APRIL 1984
Feature: Commodore SX-64. Also: Report on Winter CES Show; COMAL; 1650 Modem; (C) T6 documentation.

TPUG MAY 1984
Feature: Vic Robot; CompuServe Vidtex; $28 Vicmodem; SuperPET supersoftware; Startrek V1; Forecasting with the 8032, part 1; (V)F1, (C)T7, (C)T8 documentation.

TPUG JUNE 1984
Feature: MSD Dual Disk Drive. Also: The beginner and the disk, part 1; Fast Save by Liz Deal; ML Monitor; Forecasting with the 8032, Part 2; SuperPET 6809 assembler, Part 1; (V) T8, (P)T7, (P)T8, (S)TD, (C)M1, (C)M5 documentation.

TPUG JULY 1984
Feature: Computer aids for the disabled. Also: TPUG COMAL course; The beginner and the disk, part 2; Forecasting with the 8032; part 3 SuperPET 6809 assembler, part 2; NOS tape translator; (V)T9, (P)T9, (C)T9, (C)M6 documentation.

TPUG AUGUST/SEPTEMBER 1984
Feature: TPUG conference presentations. Also: TPUG COMAL course, part 2; The beginner and the disk, part 3; Menu handling, part 1; SuperPET 6809 assembler, part 3; (C)M7, (C)M9, (C)MA, (C)MB, (C)SD, (V)TA, (S)TE, (P)TA documentation.

TPUG OCTOBER 1984
Feature: Laser art and technology. Also: Relative files (Chris Bennett); The beginner and the disk, part 4; Menu handling, part 2; TPUG COMAL course, part 3; SuperPET 6809 Asembler, part 4; Using the user port, part 1; Forecasting with the 8032, part 4.

TPUG NOVEMBER 1984
Feature: The C-16 and the Plus/4. Also: Software theft (Jim Butterfield); Software reviews; Buscard 2 review; (C)U2, (C)U3, (C)TB, (P)TB, (S)TH documentation.

TPUG DECEMBER 1984
Feature: TPUG columns return; The beginner and the disk, part 5; COMAL course, part 4; Menu handling, part 3; Using the user port, part 2; Forecasting with the 8032, part 5; SuperPET software; (V)TB, (C)S4, (P)TC, (S)TI documentation.

TPUG JANUARY 1985
Sorry — SOLD OUT.

TPUG FEBRUARY 1985
Feature: World Of Commodore II report. Also: Foolproofing your 4040 drive; Forecasting with the 8032, Part 6; Super-0S/9 — the ultimate SuperPET operating system; Save/Load type-in program; (V)TD, (C)TD, (C)TE, (C)S5, (C)S8, (C)MC, (C)MD, (P)TE, (S)TK documentation.

TPUG MARCH 1985
Feature: Commodore 64 chess showdown. Also: Computer comfort (Jim Butterfield); Winter CES 1985; Interview with Dave Neale; What is a database?; (V)TE, (C)TF, (P)TF, (S)TL documentation.

TPUG APRIL 1985
Feature: Star Trekking and computer simulations. Also: Memory maps (Jim Butterfield); A beginner's BBS guide (Ian Wright); VIC 20 custom characters; 1985 TPUG conference; Super-OS/9 similar to Unix; (C)TG, (C)F1, (P)TG, (V)TG, (S)TM documentation.

TPUG MAY 1985
Feature: The world of TPUG. Also: In praise of the VIC 20 (Jim Butterfield); Disk errors; A beginner's BBS guide, part 2; (C)TH, (C)S9, (C)SD, (C)$1, (C)H1, (C)H4, (V)TH, (S)TG, (S)TN, (P)TH documentation.

TPUG JUNE/JULY 1985
Feature: Writing your own adventure (Steven Darnold). Also: BASIC 7.0 keywords; Exploring the C-128; A beginner's BBS guide, part 3; Using Koala pictures in programs; (C)TI, (P)TI documentation.

TPUG AUGUST/SEPTEMBER 1985
Feature: The amazing Amiga. Also: Spaghetti code (Jim Butterfield); Spooling on the B-128 (Liz Deal); A beginners BBS guide, part 4; Programmable characters; (V)TJ, (P)TJ, (P)TK, (S)TP, (S)TQ, (S)TR, (S)TS, (S)TU, (S)TV, (S)TW, (S)TX (S)TY, (C)TJ, (C)TK documentation.

TPUG OCTOBER 1985
Feature: Telecommunications. Also: C-12 music (Jim Butterfield); C1 protocol, part (Steve Punter); Wireless computing, part 1 (C)TL, (P)TL, (V)TK, (S)TK, (S)T documentation.

TPUG NOVEMBER 1985
Feature: Computer music. Also: C-128 M (Jim Butterfield); Wireless computing, par 2; C1 Protocol, part 2 (Steve Punter) Revival of the SuperPET; (V)TL, (P)TM (C)TM documentation.

TPUG DECEMBER 1985
Feature: C-64 and C-128 Graphics (Jin Butterfield). Also: Drawing with the Amig mouse; Amiga pull-out reference; BASIC/0 under Super-OS/9; (C)TN, (P)TN, (C)S documentation.

TPUG JANUARY/FEBRUARY 1986
Feature: Artificial intelligence (Dougla Hofstadter interview, etc). Also: Punter BB commands pull-out reference; Amiga dis patches; (P)TP, (V)G2, (V)TP, (B) documentation.

TPUG ISSUE 21
Feature: Exploring the C-128. Also: Gettin started with CP/M; More on C-64 music; C 128 pull-out memory map (Paul Blair) (Y)AA, (C)TQ, (V)TQ documentation.

TPUG ISSUE 22
Feature: Computers in education. Als Commodore magazines reviewed; 512K ex pansion for the C-64; Educational softwar pull-out section; Amiga screen magic (Chri Johnson); (Z)AA, (Z)AB documentation.

TPUG ISSUE 23
Feature: computer languages. Also: Lay man's guide to the 1571 burst mode (Miklo Garamszeghy); Super-OS/9 bug repor (Avygdor Moise); ESCape G 2 (Adan Herst); (Z)AC, (S)TZ, (Y)AB, (V)TR, ol SuperPET disks documentation.

The Magazine For ALL Commodore Computer Users
TPUG Magazine

TO ORDER: Indicate Month/Year of Issue
Each issue is $3.00 plus shipping

ISSUE (MONTH/YEAR OR #)	COST
If you prefer not to cut page or if you need more room you may order on a separate sheet of paper. Shipping Cost	
TOTAL	

Shipping Charges: 1–3 Magazine add $1.00
4 or more add $2.00

Name: ______
Address: ______
City/Town/P.O. ______ Postal Code ______
Prov:/State ______

COMPLETE THIS ORDER FORM AND MAIL TO:
TPUG INC. 101 DUNCAN MILL ROAD, SUITE G7 DON MILLS, ONTARIO CANADA M3B 1Z3

PAYMENT ☐ CHEQUE/MONEY ORDER; ☐ MASTER CARD ☐ VISA
CARD # ______ EXP. DATE ______
Signature ______ TPUG Membership # ______

TPUG Magazine

Publisher: Bruce Hampson
Editor: Nick Sullivan
Assistant Editors: Tim Grantham
Adam Herst
Production Manager: Astrid Kumas
Editorial/Production Assistant: Iwona Sukiennik
Advertising Sales: Bruce Hampson
Cover Illustration: Thom K. Wu
Creative Direction (Cover): Steve MacDowall
Cover Scan and Assembly: LK Graphics
Typesetting: Noesis, Toronto
Printed in Canada by: Delta Web Graphics
Scarborough, Ontario

TPUG Magazine is published 10 times a year by TPUG Inc. All rights to material published in TPUG Magazine are reserved by TPUG Inc., and no material may be reprinted without written permission except where specifically stated.

Correspondence: Send change of address and subscription enquiries to: TPUG Inc., Address Changes, 101 Duncan Mill Road, Suite G7, Don Mills, ON, Canada M3B 1Z3. TPUG Magazine welcomes freelance contributions on all aspects of Commodore computing. Contributions should be sent on disk, though accompanying hardcopy is welcome. Be sure to include return postage if you wish materials returned. Please indicate on the disk label which Commodore disk format and word processing program you have used. Payment for articles published is $30.00 per page if the author retains the copyright, and $40.00 per page if the copyright is assigned to TPUG Magazine. Payment is made on publication. All contributions are subject to editing for length and readability. Address editorial contributions and related correspondence to: The Editors, TPUG Magazine, 101 Duncan Mill Road, Suite G7, Don Mills, ON, Canada M3B 1Z3.

Circulation: 20,000

ISSN #0825-0367

VIC 20, Commodore 64 and SuperPET are trademarks of Commodore Electronics Ltd. PET is a registered trademark of Commodore Business Machines Inc. CBM is a registered trademark of Commodore Electronics Ltd.

Subscriptions to TPUG Magazine may be obtained by joining the Toronto PET Users' Group (TPUG) Inc.

Regular member (attends meetings)	$35.00 Cdn.
Student (full-time, attends meetings)	$25.00 Cdn.
Associate (Canada)	$25.00 Cdn.
Associate (U.S.)	$25.00 U.S.
	$30.00 Cdn.
Associate (Overseas — sea mail)	$35.00 U.S.
Associate (Overseas — air mail)	$45.00 U.S.

For further membership information, please contact:

TPUG Inc. Membership Information	
101 Duncan Mill Road	1552 Hertel Avenue
Suite G7	Suite 144
Don Mills, ON	Buffalo, NY
Canada M3B 1Z3	USA 14216

TPUG Telephone Numbers
Business Office (416)-445-4524
Magazine Office (416)-445-4524
Advertising Sales (416)-445-4524

DIRECTORY

86 TPUG Magazine #25

Inside Information

Flight simulators, in particular subLOGIC's versions, have been instrumental in establishing the credibility of the microcomputer. The debut of any new machine is greeted with "Does it run **Flight Simulator**?" as often as "Does it run **Lotus 1-2-3**?". There is no doubt that, for many people, the C-64 version alone justified the purchase of the system.

In this issue, you will find no less than four articles by Jim Butterfield about **Flight Simulator II**. Jim has spent many hours flying the user-friendly skies, and shares his joys, insights, and frustrations with this remarkable software. In addition, Ken Tucker outlines the fundamentals of the mathematics used in all flight simulators.

To our chagrin, serious flaws have appeared in two hardware projects published recently by *TPUG Magazine*. Colin Haig, a member of TPUG and owner of Terran Technologies, has pointed out a number of errors in the schematic diagram of the *General Purpose Environmental Device* presented in Malcolm Macarthur's article 'Personal computers and the handicapped' (Issue #22):

- The transistor should be a 2N2222, *not* 2N222.
- The diode should be a 1N914, *not* 1NB14. In fact, for better circuit protection, a 1N4002 should be used.
- The integrated circuit, U1, should be a 74ALS573. In addition to the pin connections shown, pins 1 and 10 should be tied to ground and pin 20 (incorrectly labelled RD in the diagram) should be connected to +5 V (pin 2 on the user port).
- The 12 VAC adaptor should be rated at 500 mA or better for safest operation.
- It is not explicitly stated that the relays K1, K2, K3, K4 and K5 should be 12 VDC coils, with 120 VAC contacts rated at 3 amps.

In addition, there are a number of typos we belatedly discovered:

- All the 'No's should be 'On's;
- The 'Pump's should be 'Amp's;
- 'RC' and 'PC' should read 'AC'.

We thank Colin for the corrections. We would like to apologize to those of you who may have encountered problems and remind those wishing to construct this interface to use proper construction procedure — you are working with full line voltages. In addition, make sure that both the software and hardware have been *separately* tested before attempting to use them together. Those who would like additional information can write to Colin at 1270 Indian Road, Mississauga, Ontario, Canada L5H 1S1.

Secondly, despite explicit instructions, the printer failed to flop the background of the printed circuit diagram shown in Ronald Byers article 'Expand your VIC' (Issue #24, page 8): both the the IC and the wires now go to the wrong connections. The correct version is on page 24 — the view is from the top side, as though the fiberglass board were invisible. Thus you are seeing a mirror image of the etched side of the board, with the IC and the wires in the correct position. The top-side and bottom-side edge-connector pads are out of alignment, but this should have no effect on construction.

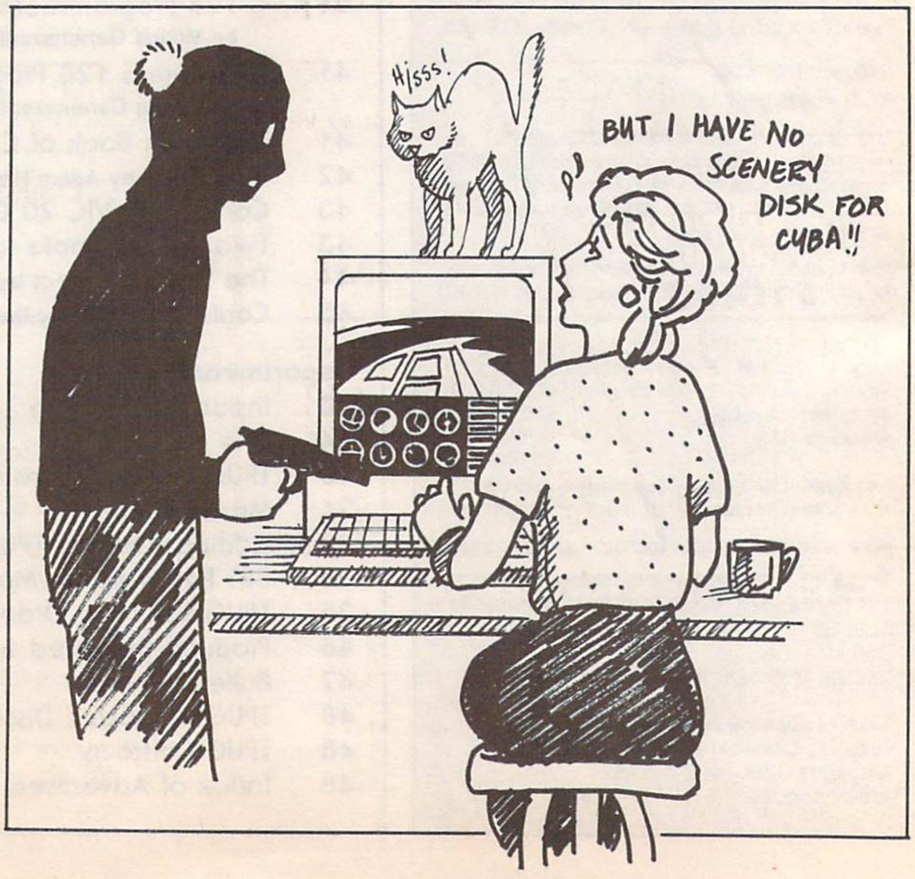
Transposition error: In the program accompanying Mike Garamszeghy's Micro Process 'Merging Program Files', there is a line 300 that actually belongs to the program found in his article 'Fun with function keys'.

We have received a wonderful disk containing arrangements of the music of Bach, complete with erudite commentary. If we have deciphered the signature correctly, these were transcribed by Eric S. Fern. This disk will be made available to the TPUG C-64 library. However, we do not have a return address, so Eric, please let us know so that we can send some software in return for your generous donation! This disk is highly recommended to fans of the Goldberg Variations.

Birth mode, part 2

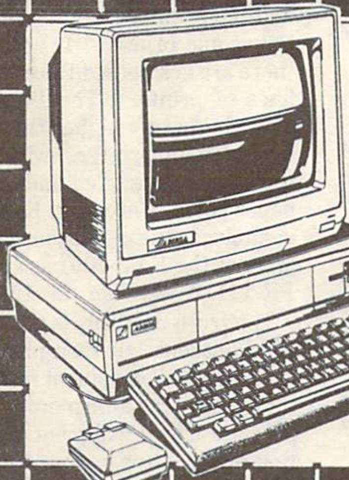
Tristan Scott Sullivan is pleased to announce his recent arrival at the home of *TPUG Magazine* editor Nick Sullivan and his wife Susan Scott. Tristan comes on board with impressive credentials, including blue eyes, plenty of brown hair, a cherubic smile, and a mass of 3681 grams. We have it on good authority that junior partners Corwin and Graham are eager to collaborate with Tristan in the development of new child/parent strategems. Our congratulations to Nick and Susan.

The Editors □



ELECTRONICS 2001

AN AUTHORIZED COMMODORE SYSTEM DEALER SINCE 1979



- **New Products constantly arriving**
- **If a product is available We have it**
- **All hardware in stock at all times**
- **Full service**

THIS MONTH'S SPECIALS

PRICES IN EFFECT UNTIL AUGUST 31,1986

SHOEBOX G/L ACCOUNTING

OUBLIETTE **$14.95**

SIDEWAYS FOR C-64 **$36.95**

IMAGE MAKER MPS 802 HIGH RES. PRINTING **$10.00**

Compuserve Starter Kit **$39.95**

PRINT SHOP COMPANION **$39.95**

NEWS ROOM **$49.95**

ICON FACTORY **$79.95**

PAPERBACK PLANNER OR FILER

128 Version Only **$39.95**

Kraft Joystick **$15.95**

$36.95

ELECTRONICS 2001 LTD.

Name	
Address	

☐ Check or money order enclosed ☐ Visa ☐ MasterCard ☐ Amex

Acct.# ____________ Exp. Date ____________ Signature ____________

5529 Yonge Street, Willowdale, Ontario M2N 5S3

Tel: (416) 229-2700

Note: All prices in Canadian Funds. Phone and mail orders welcome.
Ontario residents add 7% sales tax.
Add 5% for shipping (minimum charge $4.00)

10% discount for TPUG members on regular priced software, accessories and magazines.

LINE NOISE

TPUG Magazine *invites you to express your views on Commodore computing by writing to:*

Line Noise
TPUG Magazine
101 Duncan Mill Road, Suite G7
Don Mills, Ontario M3B 1Z3
Canada

Magnum Load and Megasoft

I am writing to you as a last resort. A year ago I purchased an MSD1 (1541 compatible?) single disk drive. It works fine for standard disk routines but is just as slow as the 1541 itself. For the past year I have been looking for a program that will speed up loading of files. After exhausting many possibilities of programs that work with the 1541 but not with the MSD drive, I came across an ad in *Compute's Gazette* for a company named Megasoft Limited. This company claimed to have a product called **Magnum Load**, a replacement Kernal ROM for the C-128 and MSD disk drives, which would load programs up to 5 times faster than normal. After ordering and receiving the ROM, I installed it and found that it would not work with my MSD drive. Every time a load was attempted, the disk drive would lock up and the computer would have to be reset. I sent this product back to Megasoft. A replacement ROM arrived last week. After installation, it was found to do the same thing as the previous one.

To make a long story short, and after seeing Megasoft starting to advertise in your magazine, I wondered if: 1) any other readers have had similar problems; 2) how reputable is this company; and, 3) is there any other product around that does work with MSD drives. Any help would be very much appreciated.

Gord Staples,
Owen Sound, Ontario

P.S. It appears that MSD has now gone out of business.

The July "86 issue of The Plattsburgh Commodore Users Group Newsletter *contains a review of **Magnum Load** by Antony Marsh, in which he also reports problems with this product. The first chip Megasoft sent him was the wrong one, and the second one did not work reliably. If anyone else is having problems with either this product or Megasoft's customer service, let us know.*

Homewriter 10

Paul Blair's article 'Dot's Nice...' (Jan/Feb 1986) is fine as far as it went, but I feel he left some things out. He did not spell out that Epson's Homewriter 10 printer *does not* support underlining, superscripts, subscripts or italics. I have tried to use it with **Bank Street Writer** and **Paperback Writer 128** and the above mentioned extras will not work.

As a neophyte in the home computer market, I bought a Homewriter mainly on Epson's reputation. I just assumed it would do everything since it was fully compatible with the C-128.

The bottom line is that I'm selling mine as soon as I can find a buyer. At the very least it should underline. Most dealers that I'm aware of (Computer Odyssey, for example) will not touch it for the previously stated reasons. I hope you will put a short note in your next issue outlining the above problems with this printer.

Kirk Girard
Dunnville, Ontario

A call to Epson Canada confirmed Kirk's information. In fairness to Paul Blair, we must point that the printer *has these capabilities — it's Epson's plug-in* interface *for Commodore computers that will not pass on the necessary escape codes, even if the word processing software can send them! This was done to ensure complete 1525 compatibility — the 1525 cannot underline or print in italics, either.*

The only way around the problem is to purchase the Centronics interface (not the C-64 interface) for the Homewriter, and connect the printer to your Commodore computer with one of the regular parallel printer interfaces, such as the Cardco G-Wiz or the Xetec Super Graphic. An awkward solution, but one that Maurice at Epson assured us would let you use the features built into the Homewriter 10.

Roland printer

I noticed two pieces of misinformation in the Roland PR-1011 printer review that were not in my submitted manuscript. My submission stated that Roland and Panasonic pinters were almost identical and never implied that Roland markets Panasonic printers! If you look closely, there are cosmetic differences in the two lines of printers. The Panasonic paper cover is slightly higher than the top surface of their printers, while the Roland top surface is angled and flush with the paper cover. Secondly, Roland PR-1011 is similar to the new Panasonic KXP-1080 and not the KXP-1091 as stated (Roland PR-1111A was like the 1091).

The truth probably is that both Roland and Panasonic buy their printers from the same original equipment manufacturer (if there ever was a misnomer?). I shall appreciate if you will print the above portion of my letter in your Line Noise section to set the facts straight.

Ranjan Bose
Winnipeg, Manitoba

C-64 coverage

I would like to bring to your attention, in the spirit of constructive criticism, the lack of information on the C-64 in *TPUG Magazine*.

May I offer to the editors a suggestion to reprint articles on the 64 that you have published in the past (for example, the C-64 memory map by Jim Butterfield)? This would spark an interest in those who have recently joined TPUG as new members, and would be of benefit to other members as well.

Ross A. Groombridge
Woodbridge, Ontario

*We feel that it is the primary responsibility of **TPUG Magazine** to provide information that is not available elsewhere, be it news about the latest products, or fresh perspectives on the current ones. Considering the variety of Commodore products, and the limited number of pages available in the magazine, we do not feel it would be in the interest of the majority of TPUG members to reprint past articles about **any** product, much less articles about the C-64, for which a vast amount of information is already available.*

*You will still find reviews of software and hardware for the C-64 in the magazine. There will be detailed coverage of the new **GEOS** operating system and* *new peripherals for the C-64. But even if we had the pages and the editorial manpower to publish more C-64 material, we would still be limited by the fact that we receive very few submissions from C-64 users like yourself. Continued support of any product requires continued input from owners — ideas, articles, programs — something that has been notably lacking among TPUG's C-64 members.*

*You are right, of course, regarding new owners and the useful information to be found in past issues. For those who do not wish to spend their hard-earned cash on the many books in print, we are making available back issues of **TPUG Magazine**. See the ad on the inside back cover for details.*

1650 modem service

I think your readers might be interested in the following information for the repair of the Commodore 1650 modem. This modem was not sold in Canada and units with problems must be returned to Commodore U.S.A. for replacement. The FCC in the States does not permit repair of modems, only replacement.

Send the 1650 modem with a $35.00 U.S. money order to:

CBM
C-2655
Westchester, Pa.
U.S.A. 19380

Put a Canada Customs sticker on the securely-wrapped package indicating 'Made in USA' and 'returned for repair'. An exchange unit will be sent to you.

Paul Gunter
Weston, Ontario

CP/M and CP

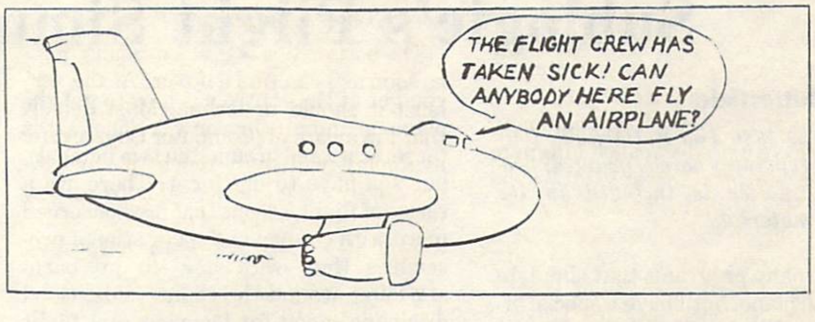
I'm on both sides of the fence as far as copy protection and copy programs is concerned. I don't mind copy protection if the vendor sells back-up copies. Some software companies don't. Sending in for a replacement copy takes too long if you are dependent on the software. Companies sometimes go out of business so a replacement is not possible. I've bought copy programs to defend myself. On the other hand, I'm a strong supporter of copyright laws. I like to encourage software companies that develop good programs. It disturbs me when people buy a computer so they can copy all the programs their friends have.

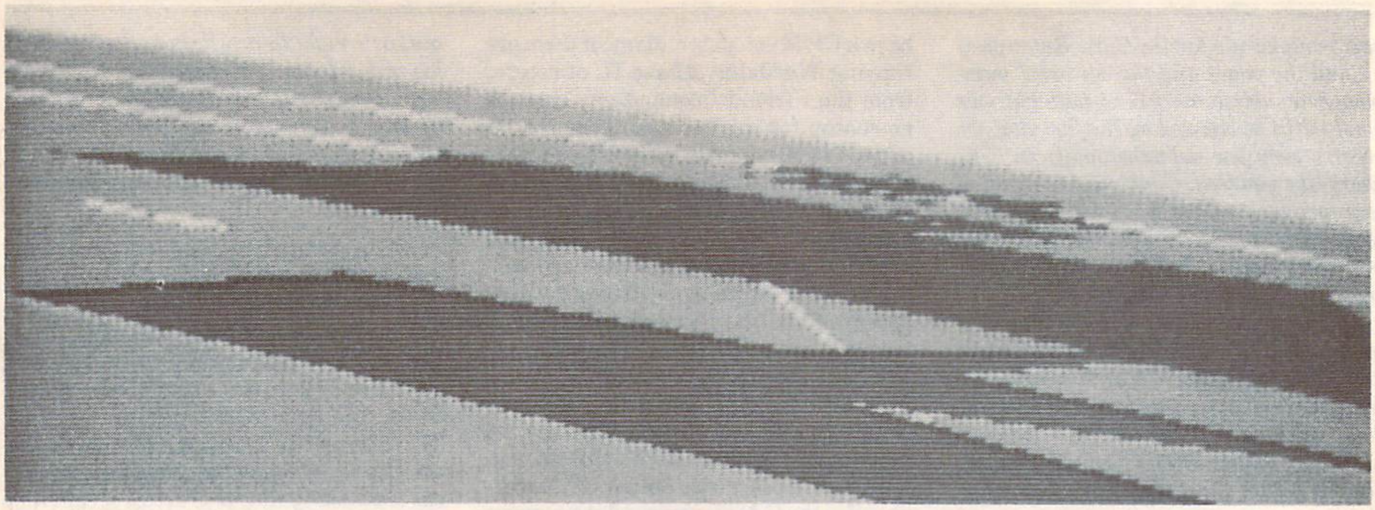
I now see the same problems with CP/M software. If a person has an old CP/M computer, it is probably all right to copy the software to C-128 format. Most people who have a C-128 do not have a CP/M computer. Many of them are copying **Wordstar**, **dBase II**, et cetera, from their friends' computers. There is no reason for this because there is a lot of good public domain software and good inexpensive CP/M software advertised in computer hobbyist and CP/M user group magazines. Computer stores that sell CP/M computers and software are willing to order anything a customer wants and even translate 8 inch disks to the 5 1/4 inch format that a C-128 can read. I've had no problem obtaining CP/M software for my C-128. The title of one article in issue #21 of *TPUG Magazine*, 'A Scrounger's Guide to CP/M: How to beg, borrow and otherwise obtain CP/M software for the C-128', surprised me because of your strong stance against stealing software.

Glynn E. Stafford, Jr.
Waldorf, Maryland

Copyrights and their enforcement through copy protection have long been a contentious issue. While a strong case can be made against copy protection, we at TPUG Magazine *are firm believers in an author's right to profit from the fruits of his or her labour. We do not, nor will we ever, condone the illegal copying of programs or documentation. By 'otherwise obtain', we did not mean 'steal'. We don't have an official stance against begging or borrowing.*

While many very good CP/M programs were available even a few short years ago, that supply has become harder to find, along with dealers who will carry these programs. Many programs are no longer produced or supported by their manufacturer, and new, legal copies are impossible to purchase. This fact, combined with the absence of copy protection, encourages the technically illegal practice of copying them. Unfortunately, this situation is unlikely to change unless CP/M machines experience a resurgence of commercial support. The only workable solution would seem to be the release of these programs as shareware or freeware. Along with the benefit to users, this could provide a source of income for the manufacturer from what would otherwise be a dead product. □


THE FLIGHT CREW HAS TAKEN SICK! CAN ANYBODY HERE FLY AN AIRPLANE?



Flying pleasure

Going aloft with Sublogic's Flight Simulator II

by Jim Butterfield

There are many programs that simulate flying an airplane, but I have a special affection for **Flight Simulator II**, by Sublogic. The thing is, it's not a game. It's real... sort of.

You're at the controls of a real plane (a Piper Cherokee Archer) and you're flying over real territory (say, San Diego to Los Angeles). The coastlines, geographic features, and runways are real ones: if you decide to land on runway 4 left at Chicago's O'Hare airport, it will be there in the simulator as it is in reality. Should you decide to fly Seattle to Olympia, it will take the same time as a real flight in the same plane.

There is no high score, no specific objective. Fly where you wish. Use instruments, or go visual. Choose summer or winter, day or night, clear weather or cloudy. On long flights, you can get slightly bored. Fly in easy mode, or choose 'reality mode'; make your plane have less than 100 per cent reliability and take your chances.

And while all the information is there in the manuals, about the only direct help you get at the controls is how to get the plane up and off the runway. The rest is there, scattered around the two manuals, but you have to dig for it. There are a series of flight lessons that are concerned more with control and navigational procedures than with how to go barnstorming around the countryside. Good discipline, good for learning real flight procedure, but the flying hacker (flapper? flacker?) may find it a little confining. If your object is at least in part to get out and enjoy the scenery, you'll find the Champaign topography a little confining. Start researching (yes, there's an index) and learn how to crash in your own style.

I have never flown a real plane, but I've cross-examined pilots as to whether or not the things I've discovered on **FS II** correspond to real flight. In the main, they do. Yes, the plane tends to 'porpoise' (oscillate up and down) if you do anything too sudden. Yes, turns become more sluggish if you put the flaps down. But I'm told that a real plane is much easier: you get extra cues, such as the feeling of motion as you turn, and the speed indication given by the sound of air flow past the cockpit. And the C-64's visual resolution is poorer than reality: it can be hard to line up on a runway.

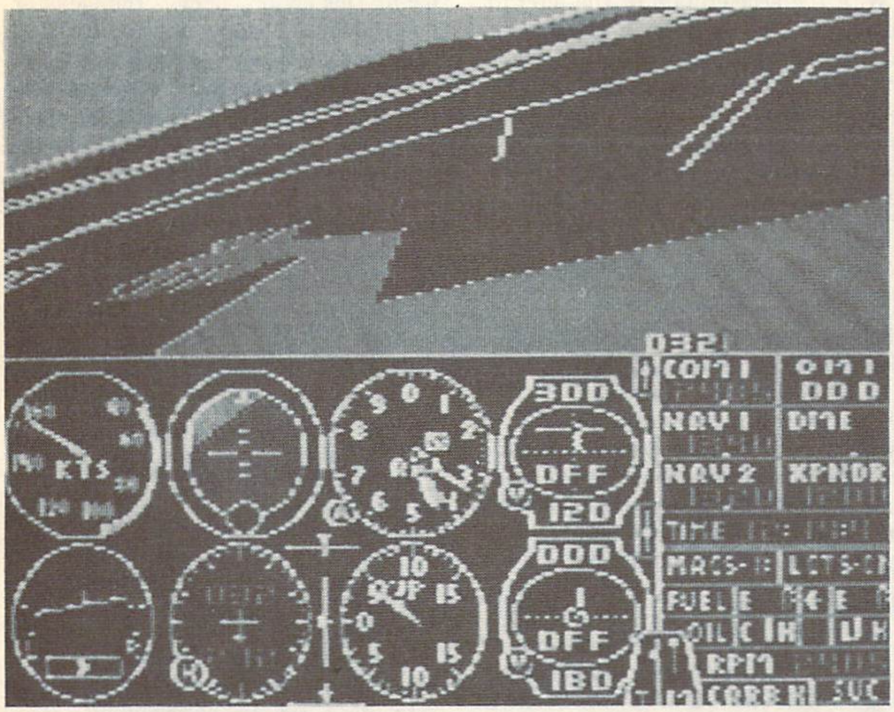
You have four scenery areas. One covers the area from Chicago to Champaign (about 150 miles); another goes from New York to Boston; a third covers San Diego to Los Angeles; and the fourth, the Puget Sound area from Olympia to north of Seattle. Most airports are shown (about 20 are typically available in each area), with accurate runway configurations. Some landmarks are included. For example, in the area of New York City, you can fly by and inspect the Empire State building, the twin towers of the World Trade Centre, the Manhattan Bridge, and the Statue of Liberty.

A couple of incidents impressed me. I was in Vancouver and, in showing **FS II**, had selected an area near Seattle. A viewer who had never seen **FS II** before exclaimed, "Hey... that's Mercer Island. Turn west and follow that highway [he identified it as I-5] and you'll come to downtown Seattle and the Space Needle!" I made the turn, and sure enough... there it was.

More recently, I was visiting my niece in Santa Cruz, California. She and her husband were going to ground school and she was actively studying the principles of flight. I was amazed to find myself explaining the purpose of an artifical horizon, and the relative merits of the gyrocompass (needs frequent resetting) as opposed to the magnetic compass (subject to short term swings and drifts). Goodness! I'd picked that and other stuff up just by roaming around on FS II.

It's fun just to cruise around. It's also interesting to in ›stigate navigation, or try your hand at finding your way somewhere. You can also set yourself goals: Can I fly across the uncharted territory from New York to Chicago? (Answer: yes, if you can figure out how to get around the fuel problem. And be prepared to yawn over four hours of dull scenery on the way).

It's something of a do-your-own-thing program. The interest level lasts much longer than for games. And it's a heckuva lower price than buying your own plane.



Jet

Jet, which is reviewed in detail in this issue by Hank Aviles, can use the same scenery as **FS II**. So if you want to fly past the Statue of Liberty at Mach 2, you may do so. It may be a good way to look over the territory — you can fly from New York to Boston in twenty minutes or so — but it's unsatisfying to use for 'sightseeing flying'.

The scenery moves by too fast, even though the screen refresh is still done at a relatively slow rate: you can be past things before you get a really good look at them. The fuel sites at the scenery airports don't work on **Jet**; fortunately, you are automatically tanked up in 'practice' mode when your fuel level drops below 10 per cent.

There is a 'zoom' feature that allows you to get a close-up view of scenery in the four directions (no diagonals in **Jet**) — that would be nice in **FS II**, especially when you're trying to line up with a runway. But this high performance aircraft goes out of control quickly if you press the wrong button, and the instruments are sparse compared to those of **FS II**. It's easy to get lost when you've inadvertently turned the wrong way, especially with no navigation aids. And it's hard to muster good control over speed and rate of descent: there are no instruments or indicators that help much other than visual judgement (and things happen *fast*!).

There's a tendency for **Jet** to fail at the instant it brings in a new batch of scenery. That was true to a lesser extent in **FS II**, especially (for some reason) in the Puget Sound area. If you touch a control at the same instant as you move into a new region, the system may give up.

Jet may be a fine chase-and-destroy game, and it's technically interesting to fly a machine so highly powered that it can accelerate going straight up. But to me it doesn't seem as rich and interesting a flying machine as good old **FS II**.

Scenery disks

'Scenery' may be the wrong word for these **FS II** extensions. They function more as aids to territory familiarization and navigation. There are no hills or mountains, no buildings, no bridges — everything is in two dimensions. You'll see rivers and lakes, coastlines and roads, and there will be lots of airports with associated navigation aids (although I've seen no *Instrument Landing Facilities*). But there's no Golden Gate Bridge, no Alcatraz or Yosemite Park, and nothing to compare to the Empire State Building as viewed in the scenery of the main program. You won't find anything to fly under, around, or between.

A typical scenery disk covers three or four regions. For example, scenery disk 3 covers the San Francisco, Los Angeles and Las Vegas areas as three separate 'scenes'. These cover all of the states of California and Nevada, overlapping slightly into Arizona. There appear to be no detail areas such as **FS II** has for selected airports or for the Statue of Liberty. No fuel is available at any airport, but it's easy to fake refuelling. When you get to the edge of one area, there will be a disk load and you (the pilot) must pull out the detail documentation for the new region. There is some overlap of navigational beacons between maps to allow you to plot a journey, but the scenery edges don't fit together too smoothly. If you fly completely out of the scenery area, you'll be put into a gross map of the USA. In principle, you can fly across the country with only one scenery disk, but you'll have to do it without navigational instruments. You can follow the main US highways; and if you want to try a flight from Toronto, start up with co-ordinates 18200N and 19248E.

The detailed documentation for each area includes a map and loose-leaf airport/navigation data. The map contains small print, some of it black on grey, and you may need perfect vision to read it. It's hard to distinguish between the digits 5 and 6 in the frequencies shown on the map. But other documentation give the information again in finer print, and great detail on runway configurations is included.

The 'Star' disks, not yet available, may contain the kind of detail suitable for sightseeing; the scenery disks don't. They do cover a great deal of territory. A flight from Chico, California, via San Francisco, Los Angeles, Las Vegas and Reno will take about eight hours. Allow suitable stops for (hypothetical) refuelling, snacks and miscellaneous other activities, and by the time you get back to Chico you'd better be ready for a night landing. □

For another perspective on ***Jet****, see Hank Aviles's review on page 36. Be sure also to read Jim's reviews in this issue of* ***40 Great Flight Simulator Adventures*** *and* ***40 More Great Flight Simulator Adventures*** *(page 37), and the Microflyte custom joystick (page 38).*

Flight simulator mathematics

by Ken Tucker

Behind the fancy graphics of flight simulator programs are complex algorithms controlling the parameters of flight. While complex, an examination of this subject should be of interest to all. This article, while trying to maintain a layman's approach, includes enough mathematical detail to allow a programmer with some knowledge of vectors and trigonometry to experiment with flight simulation programming.

A flight simulator program may be considered as two separate sections: one section for parameter calculation and another for visual representation. The aerodynamic calculations software determines the aircraft's position, speed, direction and orientation. The cockpit view software then uses this data for determining the appearance of the environment, and the instrumentation read-outs.

Flight simulation

The primary forces operating on an aircraft are *gravity, lift, thrust* and *drag.* By adding the magnitudes and directions of these forces, a single net force is obtained. We can simplify the math involved by setting the weight of the aircraft to one, and by considering force and acceleration to be the same thing. The net acceleration is the vector sum:

```
A = G+L+T+D
```

This could be accomplished in a program through the use of arrays, storing the X-axis component of A in A(1) and using A(2) and A(3) for the Y and Z axes (we will take the three axes as respectively corresponding to longitude, latitude and altitude).

The other 4 vectors, each with 3 components, can similarly be represented, and the sum would become,

```
for n=1 to 3
A(n)=G(n)+T(n)+D(n)
next n
```

As an example, consider an aircraft while it is motionless on the runway. The force of gravity downward is equal in magnitude but opposite in direction to the lift achieved by the wheels. The state of the airplane can be described as:

```
G = -1k
L = 1k
T = D = 0
A = 0
```

In our programmers' notation this is represented by:

```
G(3)=-1: L(3)=1
```

All other array elements are equal to 0.

With the acceleration of the aircraft known, it is very simple to determine the appropriate velocity (speed and direction) and position with a computer. The process by which this is accomplished is known as integration. When a force (acceleration) acts for a period of time, the velocity changes and, while the aircraft is moving, the position is changed:

```
V = V+At
P = P+Vt
```

where V is the velocity, t is the time increment, and P is the position.

In a computer program the time increment (t) can be of any duration, but is usually set so that the program runs in real time. This is accomplished by determining how long it takes the computer to complete one loop through the equations and setting t to that duration.

Wing forces

A wing produces lift through the movement of air, mainly over the top surface of the wing, resulting in upward suction. Actual forces are usually determined by experimental means in a wind tunnel. Recently, computers have been able to achieve theoretically calculated results that approximate experimentally derived results very closely.

Lift actually depends on such broad considerations as wing shape, size and texture, humidity and air density, but air speed and angle of attack are the most important. The greater the angle of attack, the bigger the 'bite' the wing takes from the air, displacing more air downward and producing more lift. All other conditions being equal, if one doubles the air speed the lift is quadrupled. Since the characteristics of the wing itself are fixed for any given aircraft we choose to simulate, we can combine the characteristics for that aircraft into a wing constant W. Then, where V is the airspeed (the magnitude of the velocity vector), and α is the angle of attack, we can derive w, the force on the wing, with:

```
w = V²Wsinα
```

provided that $-10° < \alpha < 18°$. If the wing takes too large a 'bite' (that is, the angle of attack falls outside the specified range), a 'stall' occurs and w becomes zero — there is no lift on the wing. The angle of attack is adjusted by the pilot moving the control column back and forth. This action changes the elevator flap which, in turn, points the aircraft's nose up or down in relation to the wind.

The force acting on the wing is mostly upward (lift), but a comparatively small backward force is also present. This is termed *drag.* The lift occurs in an upward direction relative to the aircraft, while the drag occurs in the direction opposite to that of the aircraft's movement. If u is the unit vector upwards relative to the aircraft's motion, and v is the unit velocity vector (V/V), then the lift L and the drag D can be determined with:

```
L = w cosα u
D ≈ -w sinα v
```

Determining drag

The result for drag in the last equation is only approximate. The fuselage and stabilizer fins, among other factors, create further drag, known as *parasitic drag.* Attempts to eliminate parasitic drag resulted in the famous flying wing experiments. A more accurate value for total drag can be obtained by incorporating a co-efficent of drag, c_d in the equation. This coefficient is determined from the real aircraft's maximum level speed at maximum thrust:

```
D = (-wsinα + cdV²)v
```

Thrust

Thrust on an aircraft results from either a propeller or a reaction engine such as a turbo jet. A jet engine's thrust magnitude is set by the pilot. The thrust direction is the direction the aircraft is pointed in. We'll assume here that the direction of flight is the direction the aircraft is pointed in. Normally this is true, but there are exceptions. In the famous aerobatic manoeuvre known as the hammer-head stall, the airplane is pointed up, while travelling down. Another famous exception is the Harrier jump jet, whose thrust direction is variable. Ignoring these exceptions, thrust becomes simply:

$T = T_v$

As an illustration, remember that the aircraft's weight is one. The actual weight of an F-104A Starfighter is about 20,000 pounds, and it has a maximum thrust of 13,000 pounds. The maximum T in this case is 13,000/20,000 or 0.65. A most familiar unit expressing this is 'g' force; at full thrust, this force is 0.65 g.

Determining the thrust of a propeller driven aircaft is more complex. The pilot is setting the horse power (Hp) of a reciprocating engine from which thrust magnitude must be derived by the formula:

$T \approx Hp/V$

From this equation it's easy to see why a propeller-driven aircraft performs better at low speeds. The lower the speed, the greater the thrust for the same power. Compare the ease with which a helicopter takes off vertically to a Harrier, which requires relatively huge engines to generate more than 1g of upward thrust for vertical take-off. In reality, of course, if the speed is zero (V = 0), thrust is not infinite, as our equation would suggest, since the propeller is still moving. One must refer to experimental data to determine thrust at full power with brakes on. A workable simulation is achieved with:

$T = Hp/(Vp+V)$

where Vp is prop speed under these conditions.

Aircraft orientation

Pitch, yaw and roll is not a new sport. It is the dance aircraft wiggle to when flying. Pitch is the up and down motion of the nose, while yaw is the left to right motion of the nose. Roll refers to rotation around the direction of motion. Of these three movements, only roll is directly controlled by the pilot. (Strictly speaking, minor changes to yaw and pitch *can* be introduced directly with the rudder and elevator respectively, but we need not take this into account). This allows considerable simplification with only minor penalty. Letting V_g be ground speed, and using θ for the pitch and ϕ the heading (which, because of our simplification, is the yaw angle), we find:

$$V_g = \sqrt{(V_x^2 + V_y^2)}$$

$$\theta = \tan^{-1}(V_z/V_g)$$

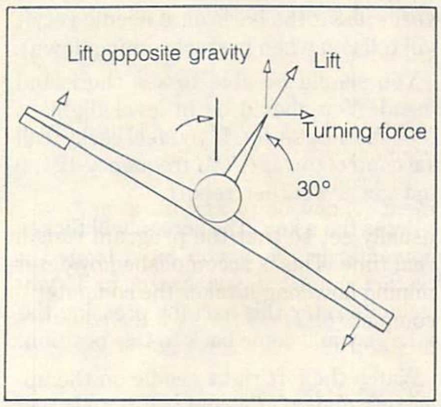
$$\phi = \tan^{-1}(V_x/V_y)$$

where V_x, V_y and V_z are the aircraft's speed along the three axes.

Most real airplanes use a steering wheel. Turning the wheel left or right causes the aircraft to roll left or right. The specific response of the aircraft to this control input is highly individual. Transportation aircraft are designed to be very stable with a comparatively large resistance to barrel rolling. Fighters however, need maximum manouverability, and are often designed unstably to this end. The result: a barrel roll at the flick of a wrist.

Turning

An aircraft flying straight and level has a lift force of equal magnitude but of opposite direction to the force of gravity. As mentioned earlier, the lift force is upwards relative to the aircraft. If the aircraft is banked (rolled 30 degrees, say) the up vector will now be pointed to the side of the aircraft, resulting in a turn (see diagram). Notice that the 'lift opposite gravity' decreases as the aircraft is banked. In real aircraft, as in



simulators, the pilot will increase the total lift by increasing the angle of attack. This keeps the 'lift opposite gravity' force constant so that no change in attitude will occur while turning.

Those readers who are mathematical masochists may enjoy confirming how our little unit 'up' vector expands to:

$$u_1 = -\sin\phi\sin\theta\cos T + \cos\phi\sin T$$

$$u_2 = \cos\phi\sin\theta\cos T - \sin\phi\sin T$$

$$u_3 = \cos\theta\cos T$$

where T is the roll angle.

Conclusion

A proper treatise on flight simulators can fill a text book. Here we have examined a 'first approximation' simulator that is realistic 95 per cent of the time. The next step (second order approximation) requires consideration of the aircraft's moments of rotation, fin stabilizing geometry and so on, to enable the simulation of spins, stalls and minor lags resulting from the difference in the direction of flight and the direction the aircraft is pointed. As the software grows more complex, the loop time increases, with the projected result becoming increasingly accurate but too slow for comfortable use. All real-time flight simulators must make a trade-off between speed and precision; the product of these two factors is a measure of the brute hardware power of the machine running the program.

When one considers the additional computer time required for the view simulator, instrumentation and possible statistical updates, the advantages of parallel processing become clear. Flight simulators have a natural affinity for parallel systems of two or three processors. One complex task, the flight simulation, passes only six numbers to the next complex task, the view simulator. With three axes of position and three angles of orientation, the view of an environment can be calculated. Without doubt, as hardware capability expands, an entirely new class of simulators will become practical and accessible to the home user for both entertainment and education. □

MA64
Version 2 of Music Assembler 64, a mini-language for music programming on the Commodore 64
Nick Sullivan

MA64 a mini-language for Music Programming on the Commodore 64

- Add music to your own programs
- Complete control of SID capabilities
- Fast, text-oriented music entry.
- Easy to learn, easier to use.
- Comprehensive Manual
- Dozens of powerful commands

29.95 CAN. 24.95 US

Ont. Residents add 7% Prov. Sales Tax
Cheque, Money Order, Visa or MasterCard
Postage & Handling Included

VISA MasterCard

WILANTA ARTS
6943 Barrisdale Dr.
Mississauga, Ont.
L5N 2H5
1-416-858-9298

Coming home on FS II

by Jim Butterfield

It's not hard to take off using Sublogic's **Flight Simulator II** — give the motor lots of power and put the nose up a bit and you'll go up. Once you're a safe distance up, you can meddle with controls (gently!) and still come out OK — keep the plane right side up, nose slightly up, enough power, and eventually the plane will sort itself out even if you can't. But as the drunk who jumped off the CN Tower said, "Flying's great, but I haven't got the trick of the landings yet."

The C-64 version of **FS II** doesn't like sudden motions, such as sharp turns, dips, or rises. If you want to bring your craft in, you must do it smoothly — don't panic. It's hard to see the runways with sufficient clarity to zip in there: the 64's resolution doesn't quite make it.

And the rules change. To descend in normal flight, you tip the nose of your plane down a bit, and down you go. But that increases your speed. So in the final stages of a landing, you must change the technique and reduce engine power in order to descend; in fact, you might tip the nose slightly up so as to reduce speed. How is a poor flying hacker to cope?

FS II has a not-well-known 'training mode' to help you on landings: the *Instrument Landing System* (ILS). It seems like an advanced feature... but the beginner can use it to help bring the plane in. Here's my recommendation.

An instrument landing 'walks you in' and teaches you a bit about descent rates, and how the screen should look as you're coming in. Mostly, take it easy. Every time you make a small adjustment, the screen will bob around a little as the plane settles into its new attitude. I'm told this particular craft is known for its 'porpoising' and can be vexing to fly. If you react too quickly, you'll exaggerate the effect and fly into a disaster.

Go to the editor and set the following values:

North position	17614
East position	22122
Altitude	2000
Heading	235
Airspeed	110
Throttle	12288
Elevators	37887

Leave the editor (press **e**). You are now 20 miles out from Martha's Vineyard, passing over Cape Cod. If you don't know your way around the instruments, press **p** to freeze everything. When you're ready, restart (press **p** again), select **n** for navigation and set frequency *NAV1* to 108.70. You'll see your distance from destination under *DME*. The upper right hand circular dial should have a left/right needle, and it should be centred. (Don't worry about the horizontal needle yet; it will tell you when to start coming down).

You should be able to see the island ahead. You should be in level flight at 2000 feet. Let it fly. If you feel cocky, call the control tower (*COM* frequency 121.4) and get a weather report.

Press the **s** key. The screen will flicker momentarily. Now you have logged a 'restart point': if you crash, or if you decide to retry this part (by pressing the **+** key), you'll come back to this position.

Watch the left/right needle on the upper left dial. If it's centred, it tells you that your position is exactly lined up with the runway. You may be flying in totally the wrong direction (although if you're on bearing 235 to 240 you're heading in), but your position is right in line. Now — if the needle drifts to left of centre, it's telling you that you're a little off the runway line. In that case, you'll want to change your course a bit to the right (say, bearing 250) to get back in line; as the needle comes back in, return to bearing 235 to 240.

So watch the needle. If it moves to right of centre, bank the plane right (gently!); if it goes to the left, bank left. As the needle comes back to centre, resume your original course of 235 to 240. You have lots of time; enjoy the view. Your altitude can drift a little; if you make a correction, do it gently, preferably using the throttle.

As you get within eight miles of the airport, watch the horizontal needle on the same dial: that's the glide slope indicator. It's pinned at the top right now, but as the glide slope comes down you'll want to follow it. Somewhere near six miles out, the glide slope pin will be about centred. Drop motor power about four notches; that will start you dropping at a rate of about 5, or 500 feet per minute. If the glide slope indicator is below centre, drop a little faster until you catch up; if it goes above the centre line, reduce your rate of descent by adding a little extra motor power.

Six miles is a magic number. Normally, you'd get a beeping sound, which is the 'outer marker', when you pass this point, but you're over water so it's not there. Tap the **n** button once to drop flaps and slow yourself down. Look for an airspeed (upper left dial) of 80 to 90 knots. Give everything time to settle. If you're going too fast, take the nose up with **b**... gently.

Your rate of descent (the dial below *ALT*) should still be about 5, for 500 feet per minute. You're controlling this on the throttle. And remember: if the glide path indicator is below the centre line, let yourself drop a bit faster (reduce power) and readjust when you catch up. You have been keeping yourself lined up left/right, haven't you? By the time you're four miles out, forget the left/right needle and fly directly toward the closest end of the runway — don't try to line up, just get to the end. Keep on the glide path.

About 0.8 miles out, you'll hear a beeping noise — the middle marker. If you're not too busy, touch **n** again for a little more flap and expect your airspeed to settle in at about 70 to 80. By now you may fly mostly visually. Keep your rate of descent at about 5. When you get to the runway, bring the nose up again by touching **b** and cut the motor as you land. Use the space bar to brake, of course. After you've landed, pick up your flaps by pressing **y** a few times.

If there's a part you can't get right, press **s** as you are going into any tricky bit during the flight to signal that you want to come back there after you crash; if you don't crash but want to try it again, press the **+** key. When you can get it down, use the editor to set up clouds at 1000 feet and come in blind. Then try a night flight.

After a few tries, you'll get a feeling for the whole landing thing. Then you can try other airports that don't have *Instrument Landing Systems*. Happy landings! □

THE COCKROACH TURBO-ROM

A plug-in replacement ROM for the 64, 128 and SX64

- **Fast LOAD, SAVE, VERIFY, FORMAT**
- Innocent SAVE with REPLACE
- Screen dump during execution
- DOS Wedge:

disk status; non-destructive directory with wild cards; scratch, rename, initialize, validate; clear directory and free BAM; change disk device number; @P/@: open 4,4:cmd4/print #4: close4; load with SHIFT/RUN-STOP; cold start, unnew; compatible with VIC switch (LOGO loads in 35 secs); has no tape or RS232 routines but don't worry; you get two ROMS in one - TURBO plus original kernal - toggle between them

$45. CDN. $35. U.S.

Cheque, money order, VISA or MASTERCARD

kobetek systems limited
1007 Commercial Street
New Minas, N.S. B4N 3C4
(902) 678-9800

Ask Someone Who Knows

If you enjoy **Jim Strasma**'s many books, and his articles in this and other magazines, you'll be glad he also edits his own highly-acclaimed computer magazine, now in its sixth year of continuous publication. Written just for owners of Commodore's many computers, each **Midnite Software Gazette** contains hundreds of brief, honest reviews.

Midnite also features timely Commodore' news, hints and articles, all organized for instant reference, and never a wasted word. Whether you are just beginning or a long-time hobbyist, each issue will help you and your computer to work together effectively.

A six issue annual subscription is $23. To subscribe, or request a sample issue, just write:

MIDNITE SOFTWARE GAZETTE
P.O. Box 1747
Champaign, IL 61820

You'll be glad you did!

```
ATTENTION COMMODORE 64/128 USERS !!!
INTRODUCING THE RAINBOW KERNAL ROM

We can now offer you an improved Kernal
ROM that replaces the stock ROM (U4) in
the C64 or C128 ! This replacement has
the following features and commands :

BUILT IN:
1.  TEN DOS WEDGE COMMAND!
2.  FUNCTION KEYS ASSIGNED TO :
    F1 ...DIRECTORY (won't erase memory)
    F2...SCREED DUMP F3...LISTS PROGRAM
    F4...RUN PROGRAM IN MEMORY F5...ENABLE
    ML MONITOR IF IN MEMORY...
    F6...CLEAR SCREEN
    F7...DELETE LINE FROM CURSOR TO END.
    F8...SYS TO ADDRESS (DEFINED BY =S)
3.  DEFAULT COLORS TO YOUR CHOICE!
4.  DEFAULT DEVICE CHANGED TO DEVICE 8!
5.  100% COMPATIBLE WITH ORGINAL ROM!
6.  100% COMPATIBLE WITH EPYX FASTLOAD!
7.  ALTERNATE FUNCTION KEYS
8.  PERSONALIZED POWER-ON MESSAGE
9.  RESET,UN-NEW,DEVICE 9 OPTION,DISABLE
    OPTION
10. + 20 MORE CONVENIENT FEATURE!
11. INSTRUCTIONS FOR INSTALLATION AND
    COMMANDS INCLUDED (older model C64's
    with a soldered U4 chip require de-
    soldering.)
INTRODUCTORY PRICE  $ 30.00 US
                      40.00 CDN-C64-VER
                      35.00 US
                      45.00 CDN-C64-MODE
                                   (128)

128 MODE VERSION COMING SOON!!!
MONEY ORDER, MASTER CARD,VISA ACCEPTED
PHONE: (604) 792-3437 FOR INFO OR ORDER
       ...2-9:00 PM.PACIFIC TIME
RAINBOW ELECTRONICS 45421 AIRPORT RD.
CHLLIWACK B.C. CANADA V2P 7K3
```

shoe box BUSINESS SOFTWARE

FAST AND EASY

30 DAY SATISFACTION GUARANTEE!
CONTINUING SOFTWARE SUPPORT!

ACCOUNTING PACKAGE

CANADIAN PAYROLL

SHOE BOX II: 97 GENERAL ACCOUNTS • FULL FINANCIAL STATEMENTS • CALCULATES TAX, DISCOUNTS, AND CCA • LARGE ADDRESS DIRECTORY • PRORATES SHARED EXPENSES AND PREPAYMENTS • TRAVEL & MEMO FILES.

CANADIAN PAYROLL: 99 EMPLOYEES • 10 PAY TYPES • 11 DEDUCTIONS • CALCULATES UIC, CPP, TAX • TAX TABLES INCLUDED • UPDATES AVAILABLE • PRINTS PAYROLL, PAY CHEQUES, PAY RECORD, T4's, EMPLOYEE LIST.

SOFTWARE INCLUDES:
SOUND CUES AND HELP KEYS, AND A VERY FRIENDLY MANUAL

FOR
COMMODORE'S 64 AND 128 COMPUTERS, COMPATIBLE DISK DRIVE, AND PRINTER

DEALERS CONTACT **PHASE 4**

Send me: SHOE BOX II ☐
CANADIAN PAYROLL ☐

Version C-64 $99.95 ________
Version C-128 129.95 ________
7% Tax (Ont. only) ________
TOTAL $ ________
Prepaid Orders Only

Name ________
Address ________
City ________ Prov. ________ Code ________

JOHN DUNLOP & ASSOCIATES LTD.
R.R. #5, ORANGEVILLE, ONTARIO L9W 2Z2 (519) 941-9572

VISA MasterCard

CHARGE MY VISA ☐ OR MASTERCARD ☐ | | | | | | | | | | | | | | | |

EXPIRATION DATE ___ | ___

Esc G 2

by Adam Herst

In this edition of **esc g 2**, the last until the fall, I want to reflect on the first year of the C-128. With three separate modes, each with its own unique characteristics, the resolution of a programming or productivity project can be accomplished in many different ways. Let's take a quick look at how the different modes are stacking up. What follows are my impressions after having worked with the 128 for close to six months.

The most used mode of the C-128 is 128 mode. This is probably because of its similarity to the C-64. If you can use a C-64 you should have no trouble starting up in 128 mode. With a similar operating system and a version of BASIC that is a superset of BASIC 2.0, many of the 64 programming tricks and techniques will work in 128 mode. While billed as a single mode, extended use has convinced me that 128 mode should really be considered as two modes: a 2 MHz, 80 column, fast mode, and a 1 MHz, 40 column, slow mode. The two modes are radically different in their characteristics and capabilities and should be used for different types of programs.

Slow mode is a graphics mode. It supports low and high resolution graphics, multicolour and sprites, all accessible through BASIC. With its 40 column screen, however, it is not particularly well-suited to text-only applications. Given this, there is absolutely no reason to use 40 column, slow mode for anything other than graphics. In contrast, the 80 column fast mode is well suited to text manipulation — *so* well suited, in fact, that the computer provides no support for graphics manipulations in this mode. Combined with its faster speed, the manipulation of large text and data files becomes effortless.

The ability of the C-128 to run the CP/M operating system with no add-on chips or boards, and the inclusion of all the necessary software in the C-128 package, undoubtedly convinced many small businesses and offices to purchase this machine. With its built in Z80 chip, 80 column screen display and multiformat disk drive, many of the problems associated with CP/M on the C-64 have been alleviated (many, not all — more on this later).

CP/M Plus, also known as CP/M 3.0, is the latest version of CP/M available. The latest version, it includes many utility programs that were considered extras in earlier versions, as well as enhancing existing programs. Next to AmigaDOS, it is probably the most powerful DOS offered with a Commodore computer. Unfortunately, this may also be the last version of CP/M. As mentioned last issue, DRI has classified CP/M as a mature product and will no longer actively support it. According to information published in *Micro/Systems Journal*, in the face of this abdication of responsibility, support for CP/M has been assumed by (forced onto?) the SIG-M CP/M users group. They will offer the same type of support that TPUG offers for the Commodore computers.

The C-128's CP/M operating system has undergone a number of changes since its release. Fortunately this is not as major an issue as bug fixes to operating systems such as Commodore DOS. Unlike changes in Commodore DOS, updates to the CP/M operating system are easily implemented on existing systems — one advantage of a disk-based system over a memory-resident system! The upgrades to the CP/M system — the most recent of which is dated December 8, 1985 — corrects important ommissions in the original release. If you haven't acquired this upgrade yet, it is available from many users' groups and online information services.

To add the new features to the upgrade systems, a rewriting of the CP/M code was required. One effect of this rewrite has been to reduce the size of the TPA (Transient Program Area) — the area of memory into which programs are loaded — from a size of 59K to 58K. This reduction, while minor, has proven to be sufficient to cause a few formerly C-128 compatible CP/M programs not to run. If you run up against this problem, your only recourse is to contact the manufacturer and plead for a rewrite of the program. If that doesn't work, you can either run the program under the old CP/M system, or abandon it.

As I said, the disk-based nature of CP/M on the 128 is a strength, in that updates and improvements are easy to implement. It is also a weakness because, like previous Commodore computers, the 128 is just not oriented towards disk intensive software. Even with the increased speed of the 128, it is the lack of speed that is the main drawback of the CP/M implementation. While the CP/M mode can run at 2 MHz, this is slow compared with most other Z80 based computers, which run the chip at anywhere from 4 to 8 MHz. This results in program operation that is noticeably, if not annoyingly, slower than on other CP/M machines.

While the recommended system consists of a single 1571, the optimal — perhaps even minimal — system required to run CP/M programs consists of two 1571s, and this configuration is expected by most CP/M programs. Unlike the programs that run on Commodore DOS and are loaded completely into memory before execution, many CP/M programs only load in portions of the program as they are needed.

However, purchasing a second 1571 may not be the only alternative for productive CP/M use. The long-awaited 1750 RAM expansion will go a long way towards improving the capabilities of the 128's CP/M mode. Firstly, it will provide a nearly 512K capacity RAM disk in CP/M mode. Support for this RAM disk is already built into the CP/M system and is available on power-up. By using this virtual drive to store either programs, data or both, disk shuffling can be drastically reduced. The addition of the memory expansion will also solve the other major problem of CP/M on the 128, the slow disk access speeds. By circumventing the slow disk-write speeds, CP/M operations that had taken minutes can now be performed in seconds. The inclusion of a RAM disk is a must in an optimal C-128 CP/M system.

Perhaps the most interesting link in the C-128 chain is the new 1571 disk drive. Its design embodies Commodore's attempts to circumvent two of the problems that were associated with the 1541 and C-64 combination. The major problem with the latter, as most CBM old-timers know, is the unbelievable (in its absence) speed with which disk access is performed. The second problem, perhaps less well known to users, was the inability of the 1541 to access the practically universal CP/M disk format, MFM.

Commodore's solution to these problems has been to include in the 1571 design a fast serial data transfer protocol, called burst mode. The first problem, snail's-pace disk access, has only been partially solved. The normal 128 operating system takes advantage of burst mode during program-file *loads* only (and very speedy loads they are too!). For saves, and sequential file accesses, however, disk access is as slow, if not slower, than with a 1541 (slower because the disk log-in procedure takes longer with a 1571 and its myriad combinations of disk formats). While burst protocol is available for these functions, they have not been included in the Kernal routines. With the presence of burst mode, fast DOS programs shouldn't be hard to write; however, it is disappointing that Commodore didn't have the foresight to include these routines in the computer.

With greater ingenuity, the second problem, MFM format compatability, has been elegantly solved. Through the use of burst mode, the 1571 is capable of reading many double density MFM-formatted disks. These include the formats of some of the most popular CP/M computers: IBM, Osborne, Epson and Kaypro. While impressive, these achievements apparently barely tap the capabilities of the 1571 disk drive. Programs already exist that allow the 1571 to format a disk with these MFM formats.

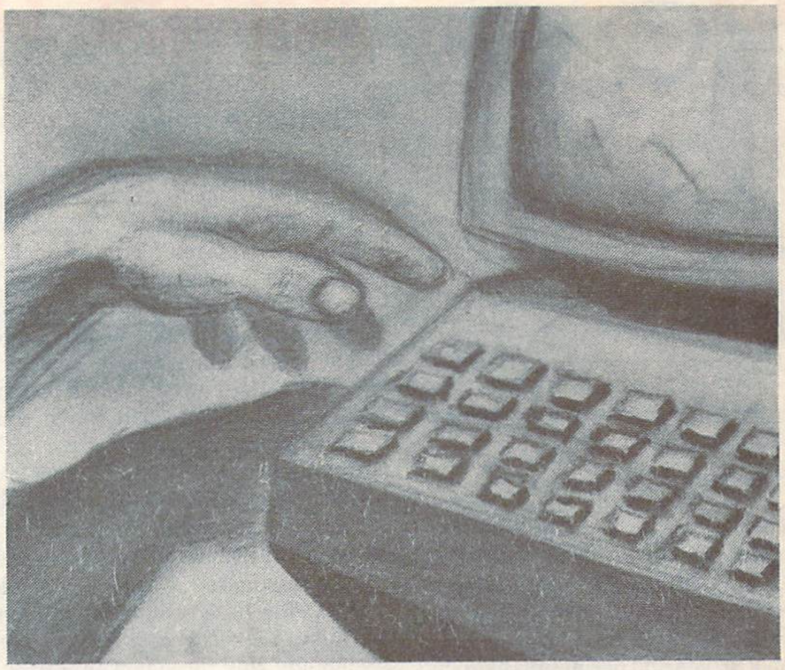
The future of the 1571 looks promising. Here at the magazine we are investigating a hardware modification that should ultimately allow the 1571 to read single density, as well as double density, disks. It also appears that the 1571 is not limited to accessing CP/M formats. Miklos Garamszeghy, a frequent contributor to *TPUG Magazine*, has recently prepared **X-Link**, a program that allows read and write access to MS-DOS formatted disks and the transfer of files to CP/M or GCR formatted disks! Look for this program to appear in the magazine and on library disks later this year.

The last mode on the 128 is the poor cousin. Intended as a 100 per cent compatible 64, it has succeeded well. In addition, many of the 128 enhancements can, with a little trickery, be accessed from 64 mode: 2 MHz clock speed, output to the 80 column screen and access to the numeric keypad, among others. Unfortunately, most of these cannot be easily used with commercial programs. In my opinion, I don't think you should be programming in 64 mode on the C-128 anyway.

Summery

It certainly is, so I'm going sailing. See you in the fall. □

TPUG PROGRAMMING CONTEST



TPUG is once again offering you the opportunity to reduce the costs of your hobby. The Librarians Committee of TPUG is sponsoring a programming contest as a means to encourage you to submit your programs to the library. The winner of this contest will be selected at random from the names of the submitters of all programs accepted by the librarians from the submissions received between the first publication date of this notice and Friday, October 31, 1986. The more programs you submit, the greater your chance of winning.

RULES

- Submissions must be received on or before the deadline.
- Submissions must be on diskette (VIC programs may be submitted on cassette — two copies, please).
- Submissions must be original material.
- Submissions can be for any Commodore machine.
- Submissions should indicate that they are contest submissions.
- All submissions become the property of TPUG.
- TPUG general policy of returning a disk of your choice on acceptance remains in effect for all submissions.
- Unaccepted disks will be returned.
- Freeware submissions will not be accepted for contest consideration.
- Submitter's name must be included in a comment statement at the start of the program as well as on the front of the disk.
- First, second and third prizes will be awarded consisting of 100, 50, 25 blank disks respectively or 25, 10, 5 disks (respectively) from the TPUG libraries.

The Librarians Committee

Amiga Dispatches

by Tim Grantham

The Toronto area is becoming a hotbed of Amiga development. Anakin Research, of Rexdale, Ontario, is doing very well with a graphics tablet/software combination called **Easyl**. This is a tool for professional artists and designers that plugs into the expansion port of the Amiga. You can use the **Easyl** software to create IFF (Interchange File Format) drawings that can then be imported into other programs; alternatively, Anakin provides drivers for all the major graphics programs, including **Deluxe Paint** and **Aegis Draw**. The tablet has a resolution of 1024 by 1024 pixels, more than enough for the Amiga and future upgrades thereof. Brad Fowles, senior design engineer for Anakin Research, told me that they are currently developing digital signal processing software that will handle both audio and video sources. Brad also told me that **Musicraft 1.1** is also being completed here in the Toronto area, although there is some doubt that Commodore will ever bring it to market.

Despite yet another quarterly loss, Commodore has something to feel good about — Amiga reportedly outsold the Atari ST during the period December to February. I hope this trend will continue with CBM's latest announcement about their marketing strategy. It seems that when the Amiga 2000 model appears, the 1000 will be dropped in price to $995 from $1295 (US). As reported in last month's column, the A2000 will come with 2 megabytes of RAM, built-in IBM compatibility, and two built-in drives. It will also have five internal sockets for expansion boards and an optional hard drive. It will be sold in the $1500 (US) range and I wouldn't be surprised if AmigaDOS is put into ROM.

Software news

John Foust, an Amiga developer and columnist for *Amazing Computers*, reports that the subLOGIC people told him that **Flight Simulator** for the Amiga would be out in August and **Jet** would be released in September... Microprose are working on a version of **Silent Service** that should be available in the fourth quarter, and Firebird (of **Elite** fame) are porting a version of **The Pawn**, complete with digitized sound and voice... Charlie Heath, author of **Txed**, announced that notices are being sent out to let current owners know how to upgrade to the recently released v1.3... TDI Software have announced a bug fix/upgrade disk for their well-received Modula-2 compiler, available for the cost of diskettes and shipping, and three new products: **Grid**, a file access utility; a disk containing Modula-2 translations of many of the ROM Kernel and Intuition routines; and a Modula-2 telecommunications package... I've had a good look at **Flow**, an idea processor, and I was not impressed. At $99.95 (US), it appears to have fewer features than Kamasoft's **OutThink** ($39.95 US) for the CP/M side of the C-128 (see Adam Herst's review elsewhere in this issue) and is considerably more expensive. Of course, **Flow** is much faster and can be multitasked with other programs, but even the speed advantage is lost if you have an expansion RAM attached to the C-128. If you need this type of program for the Amiga, check out **Txed**, which sells for $39.95 (US). I understand you can configure it to do most of what **Flow** does, and have a full-featured text editor to boot.

A company called Taurus in England has ported their **Acquisition** database management program to the Amiga. The program reportedly has full **dBASE III** compatibility, takes full advantage of **Intuition**, includes a compiler, can file data in IFF, will speak entries if desired, and has an approximation feature for when the user is unsure of the correct spelling of a search field. They claim that negotiations with interested distributors are delaying release of the product... Michael Reichmann of Batteries Included has announced that BI's **Isgur Portfolio System** will ship later this summer. He also announced the imminent arrival of **BTS, The Spreadsheet**.

Commodore's **Myndwalker** video adventure game for the Amiga is receiving rave reviews and, after seeing a beta version, I think the raves are probably justified. However, some users are experiencing a problem with 'broken sprites'. It seems that during the brain tissue part of the game, some machines will not display the **Myndwalker** character, and one of the viruses is smeared from the top of the display to the bottom. Larry Phillips of ICUG (International Commodore Users Group) in Vancouver has found a work-around for the problem: simply use the screen positioning gadget in **Preferences** to pull the display to the lower right-hand corner. Steve Ahlstrom, an Amigaforum sysop, says that there is a bug in 1.1 of the OS (Operating System) that allows sprites to be positioned outside of their hardware limits. This has been fixed in 1.2, but may be the cause of the problem.

CES (Consumer Electronics Show) in Chicago saw the announcement of many exciting products. Progressive Peripherals were showing versions of **Superbase** and **Logistix**, an integrated productivity program containing a database, timesheet and spreadsheet, complete with graphics. Electronic Arts is now shipping **Instant Music** and **Deluxe Video Construction Set; Marble Madness, Return To Atlantis** and **Adventure Construction Set** will follow shortly. **Instant Music** is a non-MIDI composition program for non-musicians that creates IFF files that can be integrated into other programs like **DCVS**. It apparently does not permit multitasking... **Maxiplan** from Maxisoft, a **Lotus**-type spreadsheet, appears to be a significantly better product than the much-trashed **Maxicomm.** It's apparently fast and efficient.

Metacomco will be releasing a BCPL compiler for the Amiga... Megasoft has released **A Filer**, a simple database program, and **A Term**, a terminal program... **Datamat**, from Transtime Technologies, mentioned in last month's column is now shipping... Mimetics MIDI interface and sequencer are on dealer's shelves in New York. They have also been showing **Soundscape**, a full-featured MIDI and sampled-sound editing program. Brad Fowles of Anakin Research told me that this is the best program of its kind, offering professional capabilities found elsewhere only on systems costing many times more... **Aegis Draw** from Aegis Development is now available. It works best if you have more than 512K RAM... Those who have seen beta versions of CBM's **AmigaTerm** have been pleased to see that it does not suffer from the slow screen output of **Online!** and other terminal programs, and can in fact keep up with 19,200 bps (bits per second)... Chang Labs' **Rags to Riches** accounting software has been substantially improved, now making full use of **Intuition** and multitasking.

Hardware

It is rumoured that the Sidecar will appear with ports for a keyboard and a monitor so that it could be used as a stand-alone unit. Also, it may have an expansion port extender, so that other Amiga peripherals can be plugged into it. Full communication between the Amiga and the Sidecar has been provided for, so that they have access to each other's files. A hard disk drive plugged into the Sidecar can be partitioned and used by both computers. While we're on the subject, it seems the **Transformer** will not work with a 68010 installed... There are persistent rumours that Commodore will be producing a DMA (Direct Memory Access) hard disk drive for the Amiga. The Tecmar and Microforge drives use software handshaking in their device-drivers and are consequently limited in their speed. A DMA drive could transfer data at typical speeds of 1.5 megabytes per second!...

The Agnes, Paula, and Denise chips are now available at $70.00 a piece, it seems... Golden Hawk's MIDI interface should be shipping now... Roger Powell has completed porting of his famous **Texture** music editing/sequencing software to the Amiga. He is now working on a bus adaptor for use with Roland's MPU-401 interface or the OpCode interface... Cardco's Amega, their 1 Meg RAM expansion unit, should be available as you read this. It costs $549.95 (US), can be stacked with other boards and peripherals, and features full auto-configuration.

Meanwhile, it's good to see that Comspec has substantially lowered the price of their 2 Meg RAM unit to $1276 (Cdn.) from $1450... SoftCircuits, Inc. has announced the arrival of their plug-in adaptor that enables the use of any standard 40 or 80 track 5 1/4 inch drive with the Amiga. The drive must have its own power supply and standard connector cable. The interface supports disk inserted/removed... Apparently as part of a cost-cutting move, some new Atari monitors are using lower-quality components, resulting in an inferior picture. Many Amiga owners had been buying the Atari monitor because it had a superior resolution than the 1080 Amiga monitor, and was substantially cheaper...

Blits and pieces

CBM appears to have fumbled with the Amiga Theatre at Expo. Seems they didn't get sufficient guarantees of exposure — Ron Troy reported that only one machine was on display! Canadian artists, though, are very grateful for the estimated $250,000 that CBM has poured into this project. It looks like the Amiga Theatre may continue after Expo 86 has finished, and the machine will take on a greater importance in its operation.

While the *Intuition* and *Hardware* manuals are now available from Addison-Wesley, the *ROM Kernel Manual* has been delayed yet again, presumably to allow for version 1.2 of **Kickstart** and **Workbench**. Addison-Wesley here in Canada say they have the *Intuition* and *Hardware* manuals in stock and ready to ship to dealers... Rockwell International's space division has purchased some 80 Amigas for their engineers and designers... The Philadelphia Phillies baseball team are using an Amiga and **Aegis Animator** to control the animation sequences on the giant screen at Veterans' Stadium... *Ami Project* is a magazine for anyone interested in programming the Amiga.

AmigaDOS 1.2

It is strongly rumoured that **Kickstart** will be put into ROM and thus will not have to be loaded from disk. It will also feature disk-caching, a more efficient directory track allocation algorithm for faster disk access, support for the 68881 math coprocessor, a **mount** command for partitioning, the ability to set the **ser:** device parameters (stop bits, x-on/x-off, parity, et cetera), and an interlace mode. This last one is interesting: I've seen two full-size (medium res) windows placed on the screen at the same time. Of course, one then has to put up with interlace flicker. There are a couple of workarounds to this problem: use **Preferences** to change the colours to ones that flicker less (I found that orange on black worked well; others suggest green on black); or do as Joe Lowery has suggested: put on a pair of sunglasses! □

WORD PROCESSOR FOR PC OFFERED BY RICH-HILL AS "USER DISTRIBUTED SOFTWARE"

First, there was SCRIPT 64. Then came SCRIPT 128. And now, there's PC SCRIPT for the PC & compatibles! It's available as "User Distributed Software", otherwise known as Shareware. You may distribute copies to other users. After trial-use, a manual is available for a modest fee. Call for further details...

731-1621

VIC 20 USERS You are not forgotten! See our line of expansion cartridges and get even more mileage from your valued unit.

731-1621

Don't junk your IEEE and parallel peripherals just yet. The necessary interface can be had from Rich-Hill Telecom International. A beta-test model has been installed at the TPUG office, and has successfully linked an 8050, 4040 and Mannesman Tally printer to the C-128 system.

731-1621

EDUCATORS...The success story behind the MULTI-LINK networking system for Commodore computers in the classroom is now history. If you're interested in networking, call for more details...

731-1621

RICH-HILL TELECOM INTERNATIONAL
200 West Beaver Creek Road
Unit 15, Richmond Hill, Ont.
Phone: (416) 731-1621

A layman's guide to burst mode

by M. Garamszeghy

The first two instalments of this four-part series on the 1571 burst mode (issues 23 and 24) covered the general syntax and application of each of the commands, and the procedure for reading data from the disk. This part describes the method for writing data in burst mode.

Part 3: Burst write

The 1571 disk drive *Burst Command Instruction Set* (BCIS) contains a single command for writing data to a disk. The *burst write* command is somewhat analogous to the standard Commodore DOS *block-write* (**b-w:** or **u2:**) command. Unfortunately, there is no *fast save* command (corresponding to the burst mode *fast load* command) that would allow you to write an entire file in burst mode.

As with most other burst mode commands, the write command will work with either MFM or GCR disks. Although the *burst write* is faster than the normal, Kernal-controlled write to the 1571, the difference is not as great as the difference in read speeds. The average speed for a *burst write*, using 256-byte sectors, is about 600 bytes per second. The corresponding figure in normal 1571 mode is about 400 bytes per second, and in 1541 mode it is about 300 bytes per second. In 1571 and burst modes, the write speeds are a factor of 3 to 5 slower than the corresponding read speeds. Unlike DOS's *block-write, burst write* can also be used to write multiple sectors in succession (up to one track's worth).

There are six basic steps to follow for a burst mode write operation. These are:

- log in the disk and send the *burst write* command string;
- set the serial port to fast output mode;
- send the data;
- set the serial port to fast input mode;
- read the burst status byte (repeat steps 2 to 5 for a multisector write);
- restore default I/O.

The easiest way to log in the disk is to use the burst mode *inquire disk* command by sending the command string:

```
"u0"+chr$(4)
```

This command will normally return a burst status byte indicating the condition on the drive controller. If you are interested in its value, the status byte can be read using the technique outlined in part 2 of this series on burst mode (*TPUG Magazine* issue 24). If you are not interested in the status byte (that is, you are sure of the type of disk in the drive) you can ignore it by sending the *burst write* command string immediately following the *inquire disk* command. As outlined in part 1 of this series, the command string for a *burst write* is:

```
"u0"+chr$(xx)+chr$(track
  #)+chr$(sector#)+chr$(
  # of sectors)+chr$(nex
  t track)
```

where **xx** can have the following values:

- 2 for a write to a GCR disk (either side) or MFM disk (side 0); stop writing if error detected
- 18 same as value 2, but for MFM disk side 1
- 66 same as value 2, but ignore errors
- 82 same as value 18, but ignore errors

As with the *burst read* command, the maximum number of sectors that can be written is equivalent to one track. The actual number depends on the disk format (for MFM disks) or the track number (for GCR disks). If you try to write more than one track's worth of sectors, you will overwrite sectors on the same track, until the number of sectors specified by the '# of sectors' parameter have been written. This could hopelessly corrupt your disk, so be careful when specifying the number of sectors to write. The 'next track' parameter is useful if you are writing several tracks in succession. Normally, before a write or read operation, the disk head will return to its 'park' position before going to the specified track and sector. If you specify the next track option, the head will go to this next track directly, without going to park first. This saves both time and wear and tear on your drive, by preventing the head from bouncing around like a yo-yo. Both the *inquire disk* and *burst write* command strings can be sent via either a BASIC **print#** statement or a machine language CHROUT routine.

The second step in the *burst write* process is to change the fast serial port direction from the default *input* mode (data flow from the 1571 to the C-128) to *output* mode (data flow from the C-128 to the 1571), and to set up the initial clock state. This is done with a short machine language routine using the new C-128 Kernal SPIN/SPOUT routine (Serial Port INput/Serial Port OUTput). To set the mode to output (SPOUT), the routine is called with the carry flag set:

```
sei
sec
jsr $ff47 ;spin/spout
lda #$40
sta clock
```

The last two instructions start the test for the system clock state on a high value. The label *clock* refers to any usable RAM location (such as zero page $fa to $ff); it is used in subsequent steps as a temporary storage location for testing the state of the system clock.

Once the system has been initialized, the data can be sent. As with the read protocol discussed in the previous instalment, data are sent to the 1571 with a simple toggle handshake using the *Acknowledge and Ready for Data* (AFRD) line:

```
      ldy #0
  ;reset buffer index
wait1 lda $dd00
    ;read AFRD clock state
      cmp $dd00
  ;debounce
      bne wait1
      eor clock
      and #$40
    ;check AFRD clock state
      beq wait1
      lda ($fa),y
    ;get RAM buffer byte
      sta $dc0c
    ;send data
      lda clock
      eor #$40
    ;toggle 'clock' state
      sta clock
wait2 lda #8
      bit $dc0d
    ;wait until byte sent
      beq wait2
      iny
      bne wait1
```

This routine assumes that a 256-byte sector of data is to be transferred (remember MFM sectors can be 128, 256, 512 or 1024 bytes, while GCR sectors written with this command are always 256 bytes. Indexing routines for other sector sizes are given in the table accompanying this article.) The first instruction resets the data buffer index. It is assumed that the data buffer address is stored in zero page locations $fa and $fb in standard low byte, high byte format. The next six instructions form a wait loop until the serial port clock pulse is in the correct phase. The next two instructions retrieve the data byte from memory and send it to the serial port. Because the size of the data buffer in bank 15 (the default bank for I/O operations) is limited (for reasons outlined in part two of this series), the **lda ($fa),y** instruction can be replaced with:

```
ldx #$3f
stx $ff00
lda ($fa),y
ldx #0
stx $ff00
```

This allows you to use most of bank 0's free RAM as a data buffer. The next group of three instructions toggles the state of the clock comparison register. The three instructions beginning with the **wait2** label form a loop until the interrupt control register (ICR) of CIA#1 signals that the transmission of the data byte is complete. The final two instructions increment the buffer pointer and repeat the process for the next byte until a complete sector has been sent.

The 1571 returns a status byte after each sector has been written. To read this byte, the fast serial port must first be set to the read (SPIN) direction followed by a ready signal to the 1571. This is done with:

```
clc
jsr $ff47 ;spin/spout
bit $dc0d ;reset CIA ICR
lda $dd00
ora #$10  ;set clock low
sta $dd00
```

The status byte can then be read with a standard burst mode read:

```
        lda #8
wait3 bit $dc0d
   ;wait for byte
        beq wait3
        lda $dc0c
   ;read status
        sta $fa
   ;store (if required)
        lda $dd00
        and #$ef
   ;set clock hi
        sta $dd00
```

If more sectors are to be written, the whole process starts over again from step 2 (set serial port to SPOUT) until the specified number of sectors has been written.

Once all sectors have been written, the final step is to restore default input/output (I/O) channels:

```
cli
jsr $ffcc ;clrchn
```

That's all there is to writing in burst mode. In the final part of this series, we will examine some of the options of the various commands in greater detail. □

Summary of Assembly Language Burst Mode Write Routines

```
General Write-a-burst-byte Routine
(used by subroutines below)

write   lda $dd00
        cmp $dd00     ;debounce clock
        bne write
        eor $0d00
        and #$40
        beq write
        ldx #$3f      ;switch to bank 0
        stx $ff00
        lda ($fa),y   ;get data
        ldx #0        ;back to bank 15
        stx $ff00
        sta $dc0c     ;send data
        lda $0d00
        eor #$40
        sta $0d00
        lda #8
wait    bit $dc0d     ;wait till sent
        beq wait
        rts
```

```
Read Status Byte
(used by subroutines below)

readst  clc
        jsr $ff47     ;set SPOUT
        bit $dc0d     ;reset ICR
        lda $dd00
        ora #$10      ;set AFRD input
        sta $dd00
        lda #8
waitr   bit $dc0d
        beq waitr     ;wait for byte
        lda $dc0c     ;get status
        sta stat      ;stash it
        lda $dd00
        and #$ef      ;reset AFRD
        sta $dd00
        rts
```

More programs overleaf . . .

Summary of Assembly Language Burst Mode Write Routines

Note: Before using any of the following routines, you must load zero page locations $fa and $fb with the low and high bytes of the start of your data buffer and call the appropriate burst mode command. Location $0d00 in the RS-232 buffer is used as temporary storage for testing the clock phase.

Write N 128 Byte Sectors

```
        ldx #number of sectors to write
        stx $fc
        ldx #0
        stx $fd      ;# sectors written
        lda #$40
        sta $0d00    ;temp storage
        sei
nexts   ldy #0
        sec
        jsr $ff47    ;set SPOUT
nextb   jsr write
        iny
        cpy #$80     ;end of sector
        bne nextb
        jsr readst   ;read status
        ldx $fd
        inx
        cpx $fc      ;last sector?
        beq end
        stx $fd
        tya
        clc
        adc #$80     ;incr pntr 128 bytes
        sta $fa
        bcc nexts    ;read next sector
        inc $fb
        jmp nexts
end     cli
        jsr $ffcc    ;clrchn
        rts
```

Write N 256 byte sectors

```
        ldx #number of sectors
        stx $fc
        ldx #0
        stx $fd      ;# sectors written
        lda #$40
        sta $0d00    ;temp storage
        sei
nexts   ldy #0
        sec
        jsr $ff47    ;set SPOUT
nextb   jsr write
        iny
        cpy #0       ;end of sector
        bne nextb
        jsr readst   ;read status
        ldx $fd
        inx
        cpx $fc      ;last sector?
        beq end
        stx $fd
        inc $fb
        jmp nexts    ;read next sector
end     cli
        jsr $ffcc
        rts
```

Write N 512 or 1024 byte sectors

```
        ldx #number of sectors
        stx $fc
        ldx #0
        stx $fd      ;# sectors written
        lda #$40
        sta $0d00    ;temp storage
        ldx #sector size/256
        stx $fe
        stx $ff
        sei
nexts   ldy #0
        sec
        jsr $ff47    ;set SPOUT
nextb   jsr write
        iny
        cpy #0       ;end of page?
        bne nextb
        ldx $fe
        dex
        stx $fe
        inc $fb
        cpx #0       ;end of sector?
        bne nextb
        jsr readst   ;read status
        ldx $ff
        stx $fe
        ldx $fd
        inx
        stx $fd
        cpx $fc      ;last sector?
        bne nexts
end     cli
        jsr $ffcc
        rts
```

A burst mode demonstration

by M. Garamszeghy

1571 burst copy is a short BASIC program with a machine language loader that uses burst read and write routines for copying disks on the 1571 drive. The program will make an exact duplicate of your GCR disks (either normal Commodore DOS or CP/M) with only two swaps for a single-sided disk or four swaps for a double-sided disk. The program is relatively fast (about 6 minutes for a single-sided disk or 12 minutes for a double-sided disk) and very easy to use — just follow the prompts on the screen. I recommend that you cover the write protect notch on the source disk to prevent disaster from striking if you accidentally mix up the disks during the copy.

While six minutes may not seem particularly fast (some 1541 disk copy programs are faster), the program is very simple (therefore reliable) and does not resort to sophisticated reprogramming of the disk drive. Nor does it blank the screen or require that extra devices be removed from the serial port. With a full disk, **1571 burst copy** is more than twice as fast as the **1571 DOS shell** disk copying utility. Even though the latter copies allocated blocks only, **1571 burst copy** will be faster for all but an almost empty disk. It is also more versatile. Because it copies everything on the disk, **1571 burst copy** can be used to copy C-128 CP/M disks (GCR format only — but it can be easily modified to copy MFM disks) and disks with unallocated random files, neither of which can be copied with the DOS shell program.

Although the target disk is formatted on both sides, only one side is used if the source disk was single-sided, thus maintaining full compatibility with the 1541 drive. □

```
10 poke48,200:clr:fori=2816to3000:readx:
   pokei,x:next
20 dimsn(35):fori=1to17:sn(i)=21:next:fo
   ri=18to24:sn(i)=19:next
30 fori=25to30:sn(i)=18:next:fori=31to35
   :sn(i)=17:next:graphicclr
40 print"<clr><2 down>***1571 burst copy
   ***":print"<2 down>by m. garamszeghy<
   2 down>"
50 gosub240:bank15:open15,8,15,"i":open8
   ,8,8,"#":ff=1:o=0
60 print#15,"u1:";8;0;18;0:print#15,"b-p
   :";8;162:get#8,il$,ih$
70 sd=1:print#15,"u1:";8;0;42;0:ifdsthen
   sd=0
80 print"<down>copying"sd+1"sides...<dow
   n>":close8:print#15,"u0"+chr$(4)
90 print"reading...":a=52:b=0:fori=1to9:
   gosub270:a=a+21:next
100 a=5:b=1:fori=10to17:gosub270:a=a+21:
    next
110 gosub260:iffthen print"<down>format
    ting...":print#15,"u0"+chr$(6)+chr$(
    0)+il$+ih$
120 print#15,"u0"+chr$(4):print"writing.
    ..":ff=0
130 a=52:b=0:fori=1to9:gosub280:a=a+21:n
    ext
140 a=5:b=1:fori=10to17:gosub280:a=a+21:
    next
150 gosub240:print#15,"u0"+chr$(4)
160 print"reading...":a=52:b=0:fori=18to
    27:gosub270:a=a+sn(i):next
170 a=5:b=1:fori=28to35:gosub270:a=a+sn(
    i):next
180 gosub260:print#15,"u0"+chr$(4):print
    "writing..."
190 a=52:b=0:fori=18to27:gosub280:a=a+sn
    (i):next
200 a=5:b=1:fori=28to35:gosub280:a=a+sn(
    i):next
210 ifsdtheno=35:gosub240:sd=0:goto80
220 print"<clr><4 down>***done***":print
    #15,"i":dclose
230 input "<2 down>copy another [y/n]";c
    a$:ifca$="y"then40:elseend
240 print"<down>insert source disk..then
    <space>press return"
250 getkeya$:ifa$<>chr$(13)then250:elser
    eturn
260 print"<down>insert target disk..then
    <space>press return":goto250
270 print#15,"u0"+chr$(64)+chr$(i+o)+chr
    $(0)+chr$(sn(i))+chr$(i+o+1):sys2928
    ,a,sn(i),b:return
280 print#15,"u0"+chr$(66)+chr$(i+o)+chr
    $(0)+chr$(sn(i))+chr$(i+o+1):sys2816
    ,a,sn(i),b:return
290 data 133,251,134,252,132,253,169,  0
    ,133,250,120,169, 64,133,254,160
300 data   0, 56, 32, 71,255,173,  0,221
    ,205,  0,221,208,248, 69,254, 41
310 data  64,240,242,166,253,169,250, 32
    ,116,255,141, 12,220,165,254, 73
320 data  64,133,254,169,  8, 44, 13,220
    ,240,251,200,208,216, 24, 32, 71
330 data 255, 44, 13,220,173,  0,221,  9
    , 16,141,  0,221,169,  8, 44, 13
340 data 220,240,251,173, 12,220,133,255
    ,173,  0,221, 41,239,141,  0,221
350 data 198,252,240,  6,230,251, 76, 17
    , 11,234, 88, 32,204,255, 96,255
360 data 133,251,134,252,132,253,160,  0
    ,132,250,120, 44, 13,220, 32,171
370 data  11, 32,164, 11, 32,164, 11,162
    ,250,142,185,  2,166,253, 32,119
380 data 255,200,208,240,198,252,240,  5
    ,230,251, 76,129, 11, 88, 32,204
390 data 255, 96,255,  0,169,  8, 44, 13
    ,220,240,251,173,  0,221, 73, 16
400 data 141,  0,221,173, 12,220, 96,  3
    ,  0
```

Dot-matrix printer basics

by Ranjan Bose

When the Commodore 64 computer was introduced several years ago, buyers had little trouble choosing among peripheral devices such as disk drives and printers. Only one brand name of compatible devices existed — CBM. In the last two years, however, several independents have launched C-64 compatible disk drives and even cassette recorders. While Commodore peripherals used to be less expensive, this is no longer true.

Printers have lagged behind in this respect. Many models of parallel printers existed, but none of them could be easily connected to the C-64. And these printers and the necessary interfaces were expensive. Commodore printers (152x and now the MPS80x series) had limited features and barely acceptable print quality, but carried very attractive price tags. In the last year or so, two things have happened on the printer front. Third party printers have become more affordable and many inexpensive C-64 interfaces have been introduced. The prices of Commodore printers have also gone down by almost 40 per cent, but the price difference between them and a third party dot matrix printer, with interface, has diminished to the point where buying a Commodore printer should not be the automatic response. Several manufacturers (Epson, Star, Blue Chip, Riteman, among others) have started selling Commodore-ready printers with built-in, Commodore-serial to Centronics-parallel interfaces. Star, for instance, sells the SL-10C model, which costs the same as MPS802 yet also does graphics, italics and NLQ (near letter quality) printing.

Why parallel?

Buying a universal, parallel printer makes sense for other reasons as well. It can be directly hooked up with other computers (like the Amiga), and the depreciation on a parallel printer is far less than on a restricted, Commodore-ready printer. Economics aside, a parallel printer typically offers a vast array of features over those offered on Commodore-ready printers. To name a few: multiple pitches; type styles like bold, emphasized, NLQ, italics, underlining, superscripts and subscripts; page formatting; horizontal and vertical tabulations; international character sets; downloadable characters; word-processing functions like margin setting, left, right (or both) justification, centering and proportional spacing; and dot-addressable graphics from 60 to 240 dots per inch. You may never have occasion to use all the features such a printer offers; however, it is always better to have more features than are necessary at the moment to leave some room for future growth.

Most interfaces will let you use your printer in a 1525 emulation mode that makes available the Commodore graphic characters and reverse field printing. With a Commodore printer you are limited to listing a program in only one way. With interfaces you can usually list a program so that the control and graphic characters are listed as mnemonics, as key presses, as ASCII codes or as graphic characters. Again, the options possibly outnumber your immediate requirements, but they are there should you need them.

Setting up

After you have purchased your dream printer and interface, you will have to connect the serial cable coming out of your interface to your computer's (or disk drive's) serial port. A thin wire ending in an adapter goes either to the Datasette port or joystick port for powering your interface. *All power switches should be off* when you plug in the adapter unless you love blowing-up fuses or even microchips! The other connection from the interface goes to the Centronics port on your printer.

The next step is to set the tiny DIP (Dual Inline Package) switches on your interface and/or your printer so that the computer, interface and printer can communicate properly. Use a sturdy toothpick for this manoeuvre. The DIP switches usually control functions like line feeds, printer type, interface mode and device number selection. Since the C-64 usually does not send a line feed with each carriage return to the printer (unless you have opened the file to the printer with a file number greater than 127), and since the printer DIP switches are usually inconvenient to access, you should select the setting that allows only the interface to send line feeds. You will thus avoid overprinting on the same line or unwanted double spacing. The manuals are explicit about these very critical settings and should be closely followed.

Your interface can work in three basic modes (DIP switch selectable or, rarely, software selectable). In the transparent mode, all data will go through the interface unaltered. This is usually used with word processors and graphics programs, and is sometimes the only way to access certain special features of your printer. The second mode is ASCII or text only. Commodore ASCII, or PETSCII (Commodore's quirky version of the otherwise almost universal ASCII code) is converted to true ASCII in this mode. The third mode is the 1525 emulation mode. Your printer essentially becomes a 1525 (except it is usually faster). All non-1525 codes are blocked by the interface. This mode is used with commercial, 1525-compatible programs.

Some interfaces (Xetec, for instance) have yet another mode, which is a combination of transparent, ASCII conversion and 1525 emulation all rolled into one. In this mode you can access all the special features of your printer and at the same time print Commodore graphic characters and print in reverse field. There are also fancy interfaces with huge buffers and multiple fonts. On the other hand, there are also lowly interfaces that support only text. If you are that hard up, you are better off buying a Commodore-ready printer or, better still, skip lunches and beer and buy a graphics interface when you have lost some weight. Most newer interfaces have active switches that immediately bring into effect changes made in the DIP settings. Earlier interfaces used to read the settings on power up; changes in the DIP settings could only be instituted before switching the power on.

Once the data signal gets past the interface, it activates the printer to print something or to perform some function. A brief description of the major printer-features follows

Text

Dot-matrix printers usually print in a field of 9 by 9 dots. By selectively printing some of these dots (typically 5 by 7), a character shape is generated. These character shapes are stored as binary code in the printer and interface ROM and, in most cases, some of these codes can be replaced or new code added to provide a custom-designed character-set. Printers usually have several character sets for printing regular characters, NLQ characters, italics, international characters, proportionally spaced characters and, sometimes, special graphic characters (usually IBM compatible). By varying the speed of travel of the printhead, the pitch of the characters can be altered to get from 10 to 17 cpi (characters per inch).

The regular characters generate what is known as draft quality printout, with each of the dots forming a character being discernible. The tiny spaces between horizontal dots in a character can be abolished by printing the same column of dots twice while the printhead is travelling at half speed (at regular speed this would produce a double width character). This is known as emphasized print. Since condensed (17 cpi) pitch already has high horizontal dot density, emphasized printing is not permitted (nor necessary) in this pitch. The tiny gaps between vertical dots can be covered by double printing, where a line is printed, the paper moved by a fraction of an inch, and the line printed again. The emphasized and double strike modes can be combined to give a very dark print. These were employed in the pre-NLQ days to improve the appearance of the printout.

Near letter quality printing uses multiple-pass printing and special letter shapes to produce print that in some printers is almost identical to typewritten copy. All these enhanced printing modes cut down the printing speed by 50 to 80 per cent and eat up the ribbon hungrily. Superscripts and subscripts are generated by printing characters so that they are compressed upwards or downwards to half their normal height. These are usually printed in two passes to improve readability.

Formatting

Many parallel printers allow formatting of a printed page. Left, right, top and bottom margins can be set, and form length can be specified in inches or by number of lines. The spacing between lines can be altered in steps of 1/72nd, 1/144th or 1/216th of an inch, depending on the printer. There are facilities for setting up horizontal tabs (as in a typewriter) or even vertical tabs. These help in the printing of tables. Some printers also support justification, centering, microjustification and more.

Graphics

In addition to printing text, most dot matrix printers also allow you to print diagrams and graphic designs. Commodore 1525 printers allow a horizontal density of 60 dots per inch and vertical density of 7 dots per line. Most other printers allow 8 or 9 pin vertical density per line and several horizontal densities ranging from 60 dpi to 240 dpi. Most of these modes are accessed by sending special character string codes following the **escape** code. On the Commodore 64, pressing **ctrl-[** in quote mode generates this code. You may also send it as **chr$(27)**.

Most printers allow you to use both single sheets or tractor-driven fanfold paper. The printers that push fanfold from behind the platen tend to bundle up and jam paper at times. Alternatively, where the tractor comes after the platen and pulls the paper, you get better flow of paper but you lose the first sheet, a minor inconvenience. Printing speeds vary from 100 to 160 cps (characters per second) for draft mode, and 20 to 40 cps for NLQ mode. Generally speaking, the faster a printer, the noisier it is. On some printers you can select half-speed printing for reduced noise (this feature could save your marriage!).

Most printers use ribbons in easily replaceable cartridges, while a few allow you to use regular typing ribbon spools. The latter method is cheaper, and easily available, but is messy to change. The ribbons themselves are made of either carbon film, which gives crisper images but has a short life (about 1 million characters, or 500 pages of double spaced text), or inked nylon (about 2 to 3 million characters). Some of these can be re-inked several times. Since the ribbon rubs over the printhead and its pins continuously, the ink contains special lubricants to reduce friction-induced wear of the printhead. *Never re-ink a ribbon with ordinary inks.*

The printhead itself has a life expectancy of over 100 million characters. Some printheads are user-replaceable while others are not. Most printers achieve increased print output by using faster line-feeding and bidirectional logic-seeking printing. This enables the printhead to print from left to right and from right to left, starting with whichever edge is closer to its current position. Since printers have so many moving parts, they are potentially prone to breakdowns. Warranties range from 90 days to two years and should be an important factor to consider at the time of purchase. ☐

Marketplace

COMPUTER RENTALS

We buy, trade, sell and repair Micro Computers

• 1541 Alignment	$ 29.00
• 4040 or 8050 Drive	$795.00
• 2031 Drive	$325.00
• SuperPET	$895.00
• Wordpro 4+ 4032/8032	$150.00
• VisiCalc	$150.00
• Manager 8032	$150.00
• IBM Compatibles from	$650.00
• Rentals from	$12/week

1262 Don Mills Road
Suite 94
Don Mills, Ontario
M3B 2W7
Tel: (416) 446-1035

CONFERENCE ANNOUNCEMENT

The TPUG annual conference will be held this year in conjunction with the World of Commodore IV show, December 4 to 7, 1986, in the International Centre in Toronto.

The World of Commodore has traditionally been the world's best showplace for Commodore's latest products. It is also one of the largest computer shows anywhere, attracting more than 40,000 people.

This conference will feature a galaxy of expert speakers, and a Saturday night banquet that you won't want to miss. As usual, there will be a flea market and used equipment sale, and a copying session where you can pick up TPUG disks at a special conference price.

Free admission to the World of Commodore is part of your ticket, but we will also be providing alternate programming for guests who might not be interested in all aspects of the show and conference. Hotels in the immediate area have been booked for your stay.

Application forms for this year's conference will be available in the next issue, but set aside the time now. This will be our best and most exciting conference ever. Be there.

This space could be advertising *your* product.

Call Lois Lackey, (416) 445-4524

Nodular Programming

by Steve Punter

I define a *BBSer* as anyone who spends time calling microcomputer-based bulletin board systems. Here in the Toronto area it's a virtual paradise for BBSers, as there can be upwards of two hundred, or more, boards operating at any given moment. Toronto BBSers have little need to contact boards outside the toll-free range of their trusty telephones.

For BBSers in smaller towns or cities, however, where there can be as few as one or two boards, calling long distance can be a fact of life. Do Toronto BBSers have it made, or is there something they are missing that small town types aren't? In my opinion there is: diversity. People in Toronto are quite different from people in Chicago, who are quite different from people in San Fransisco.

The phenomenon of mass communications has, admittedly, made us a lot more alike, but each city has its own set of concerns and its own viewpoints. By restricting our conversations to those living in the same region, we effectively limit our experiences and close off avenues of discovery. Of course, if this is all there was to it, we'd all run to our computers right now and call exotic far away places, but we all know why we don't: it costs too much.

For example, calls to the Los Angeles area are over one dollar a minute from Toronto during peak periods. We could wait until the evening and get one third off or we could become night owls and call after midnight when the rates are two thirds off. Even then, a 15 minute call to a board in Los Angeles would cost nearly $5.25. Were we to do that too often, the costs would be enormous. And for what? The opportunity of hearing a slightly different point of view?

Even the beginners amongst us know that a better way is to get accounts on such nationwide mainframe systems such as CompuServe or Delphi. We can meet all the people we like there, and they will most certainly be from exotic, far away, places. But these service cost a lot of money too, as anyone who has spent much time on one will tell you.

A much better way to approach the whole problem would be an honest-to-god *electronic mail* system where, as with regular mail, a fixed price is paid to send a *letter* to someone in a far away place.

A marvellous vehicle for this sort of operation would be a free-access, microcomputer-based bulletin board system. Do such systems exist? The answer to that is a resounding yes!

What is PunterNet

To any casual caller, there is no apparent difference between a non-PunterNet **BBS64** and a PunterNet equipped **BBS64**. Upon closer examination, though, you will find that some of the message headers have a new look. For example, a standard message header is a lot like this:

```
Msg # : 165 — Ref 7009
From  : STEVE PUNTER
To    : ALL TPUG MEMBERS
Posted : 1422h on 16-May-86 * TPUG
Subject: A Regular Message
```

On the other hand, a message that was received over PunterNet will look a lot like this:

```
Msg # : 166 — Ref 7010
From  : STEVE PUNTER (+1)
To    : ALL TPUG MEMBERS
Rec'd  : 1422h on 16-May-86 * TPUG
Subject: A PunterNet (tm) Message
Mailed : 1202h on 15-May-86 * PSI
```

Three main differences can be noted. The first is the presence of the **(+1)** at the end of the sender's name. This information tells the reader which *node* of PunterNet the message came from.

What is a 'node'? Well, PunterNet is made up of a group of **BBS64** bulletin boards, each an independent system in its own right, but capable of something other Commodore boards are not: it talk with one other PunterNet boards, automatically, without coordination from its sysop (system operator). This is the key to the entire networking concept. I'll go into further detail on this later. For now, you should know that each of PunterNet board is assigned a number. This is its *node number*. In the above example, the **(+1)** told us that the message came from Node 1.

The second difference concerns the label on the fourth line of the header. Rather than saying *Posted*, this line now reads *Rec'd* (short for 'Received'), since it's not accurate to call this piece of data the posted date. This date is the actual time which the message came in over the network, and has nothing to do with when it was originally posted.

The third difference you will note is the addition of the separate, sixth line, which *does* report when the message was actually posted. As you can see, our example was posted almost 26 hours before it was received, so there is clearly more going on here than meets the eye.

Tale of a message

A PunterNet message starts life like any other message on whatever board it is being written. After the sender signs off, though, the path to its recipient's eyes begins. First off, it is deleted from the message base of the sending node, and added to a special file along with all the other node messages. You can think of this file as a mailbox.

Since telephone rates are cheapest during the wee hours of the morning, it would be foolish for one node to contact another at any other time. It would also be foolish to contact another node right away, since there might be other messages waiting to go out to the same place. Think of this as not sending the mail truck around to the mailbox every time someone drops something in it.

Eventually though, the mailbox will get full enough, or sufficient time will have elapsed, that it will become necessary to transmit the message to its destination. At that time, the sending node will start making attempts to contact the destination node. Contact will sooner or later be made, and the messages will then be transferred under my C1 communications protocol, thus assuring error free transmission.

The destination node now adds the new mail to its own message base, ready for local callers to see (except for private messages, which are readable only by the sysop and the addressee). This completes the process by which a message composed by a local caller in a far away city is made available to be read locally by callers to the destination node. The only cost is the long distance call from the node of origin to the destination node. Most nodes operate at 1200 baud, and can transmit all the required data very quickly.

Costs will vary, depending upon the locations of the two nodes and the size of the data to be transmitted, but overall the amounts are extremely low, allowing very reasonable *postage rates*. Early experience with the network has suggested that the current cost of about 30 cents per message may be a little low, but is fairly close to a good, realistic value. Very few people would argue that even 35 cents isn't a good deal.

Who? Where? Why?

Let's start with *Where?* PunterNet is quite young, and there is not yet the continent-wide coverage that there may one day be. Nonetheless, there are already 25 active nodes (see box).

Who and *why* go hand in hand. Whom you talk to and why are most definitely up to you. I've encountered quite a bit of resistance to joining the net from users who cannot conceive of any purpose to the whole exercise: PunterNet is a new venture, and it will probably take the average BBSer a while to see what use there is in it all.

Node sysops have thus far been the prime users of the network, yet beyond that a definite pattern is developing. Users of nodes in cities with few boards are jumping on the concept wholeheartedly, while users in saturated areas, such as Toronto, are showing great reluctance. This is a shame since it's precisely that group of people the network will benefit most.

Delving deeper

PunterNet messages are sent in the same way as messages on other systems, except in the way you enter the recipient's name. For example, should you wish to send a message to me on the TPUG board, while signed onto the TPUG board, you'd merely address the message to **steve punter**. On the other hand, were you signed onto TPUG and wished to send a message to me on *my* board (PSI-WordPro), you would address the message to **1/steve punter**.

The key to the above entry is the 1/, which tells the TPUG board to direct this message to Node 1, rather than merely putting it up in the message base. Node numbers are easily determined on any PunterNet board by entering the Bulletin Section and calling for a file called **nodes**.

In the Toronto area, we are fortunate to have about 10 local nodes, amongst which messages may be exchanged at no charge. Of course, a small charge must be levied for messages sent to long distance nodes, since the board operator must recover the long distance charges incurred. This is where PunterNet gives you the edge, since at about 30 to 40 cents for any long distance message, you come out ahead of any other message transfer method available.

Charges for messages are made on the basis of a simple formula: a fixed number of cents per formatted line with a minimum charge. Just about all the nodes use my suggested charge structure of one cent per line with a 30 cent minimum. As mentioned before, local Net messages are free of charge. To help you determine how much a message is going to cost before you begin to enter it, the cents per line and minimum charge are presented to you just prior to you typing the first letter. Once the message has been completed and sent, you will be told of the total cost.

Messages are paid for by deducting the total cost of each Net message from an account you buy from the sysop of your local PunterNet node. How large an account you purchase will depend on how much you expect to use the network.

What happens now?

Now that you have signed off, **BBS64** identifies all the PunterNet messages you have sent and moves them into a file called **outfile**, then deletes the original messages from the message base. **Outfile** is like a mailbox, and will contain messages destined for many different nodes. When your message is actually transmitted will depend on a number of factors. First, if the message was destined for a local node, it will be transmitted that same night, with only a system failure at either end holding it up. Long distance messages are another matter. Since there is a minimum charge levied by the phone company for any long distance call, it would be foolish to send the message immediately. Instead, the board will wait for the accumulation of at least four messages for the same node. Of course, a seldom used node could conceivably never accumulate four messages and thus cause yours to remain in limbo. For this reason, any node messages will be sent once they become four days old.

The calling of other nodes occurs between the hours of midnight and 7 am (local time), when long distance rates are much cheaper. To guarantee that each node is called a sufficient number of times to get through, the board will call each node for which it has mail after any user signs off. On seldom used boards this approach may not be good enough, so a random time each hour is selected in which to make the calls. In addition, to help get through to boards that get huge volumes of callers even in the wee hours of the morning, a *window* of 1 hour and 15 minutes is opened each night when only nodes may call in. This window is designed to appear at different local times in each time zone, thus causing the window to open simultaneously, all across North America.

As each node is called, the messages destined for it are collected together into a single file. Once a connection has been made to the destination node, this file is transmitted in its entirety using my C1 protocol. After transmission is complete, the calling board hangs up and goes on to the next node it must call (if any).

In the meantime, over at the destination node, gears start to turn (so to speak). First, the received data is appended to the end of a file called **infile**, but not until it's been processed to remove any possibly corrupted data. This is to minimize possible damage in the event that an intruder has managed to infiltrate the system.

The node now checks if there is enough space in it's own message base to store the incoming messages. If so, the messages in the **infile** are transferred there, ready to be read by the next caller. If space runs out, the remaining network messages are left in the **infile** where they continue to build up as other nodes call in. Should a user delete messages that open up enough space for even one of these pending message to be brought in, the board will do so right after that user signs off. As you can see, the process is completely automatic, and is designed to cope with most eventualities.

Local long distance

In certain circumstances, usually around metropolitan areas such as Toronto, a node might be just outside the local range of most nodes. I say most, because sometimes one node is so located that it is local to all nodes. In the Toronto area, for instance, Georgetown, while long distance from Toronto, is not long distance from Brampton. Brampton, in turn, is not long distance from Toronto, and provides an opportunity to put some fancy computing power to work.

As it turns out, there *is* a Brampton node (#12), allowing a process known as *redirection* to be implemented. Since the Georgetown node (#10) can only call Toronto long distance, the sysop *redirects* all messages destined for any of the Toronto nodes via Node 12 in Brampton. In the meantime, the Toronto nodes redirect all their messages destined for Node 10 through Node 12 as well.

What happens is this (we'll use the ex-

Continued overleaf...

ample of sending a message from Node 10 in Georgetown to Node 7 in Toronto): a user enters a message on Node 10 addressed to **7/all**. His board checks its information on Node 7 and notices that it's been redirected via Node 12, and so checks its information on that node as well. Noting that Node 12 is a local call, the user is informed that his message will be free of charge. When the user signs off, his network message is placed in the **outfile** mailbox as usual, but is marked for delivery to Node 12, not Node 7.

During the night, Node 10 discovers it has data to send to Node 12 (a local node) and does so. Upon arrival at Node 12, the message is identified as a transient and placed into the **outfile** destined for Node 7. Since Node 7 is a local node, attempts are made to call it that same night. The result of all of this is a completely automatic system that allows certain long distance barriers to be broken.

Redirection is not limited to a single jump, and can be stretched out to any length, although the whole process gets nightmarishly complicated. Nevertheless, you can't jump through a location which doesn't have a node, and you certainly can't make jumps all across the country. It is impossible to call from one area code to another and not incur charges, no matter how close the two nodes are, so the process is limited to within a single area code.

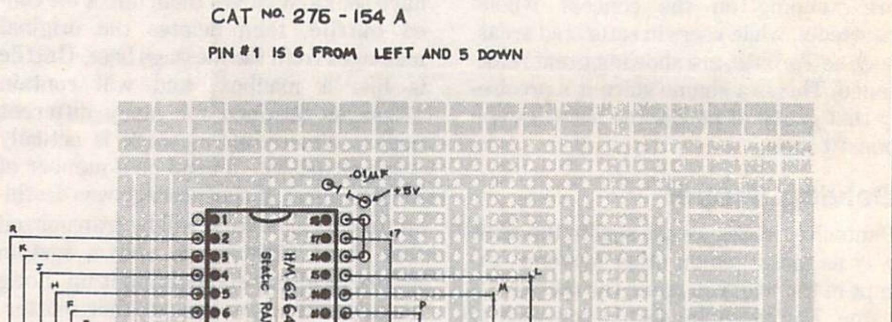
Another good use for redirection is to allow long distance nodes to *pool* all their messages for a larger metropolitan area with more than one node (like Toronto). Of course, sysops should contact the node they wish to use for redirection before going ahead with something like this since, should the board they choose not have its redirection flag set, all messages sent will be lost.

PunterNet is still rather young and has much growing to do. Part of that growth will depend on the level of participation. With luck and effort, PunterNet will eventually provide a low-cost, user-maintained, continent-wide message service. □

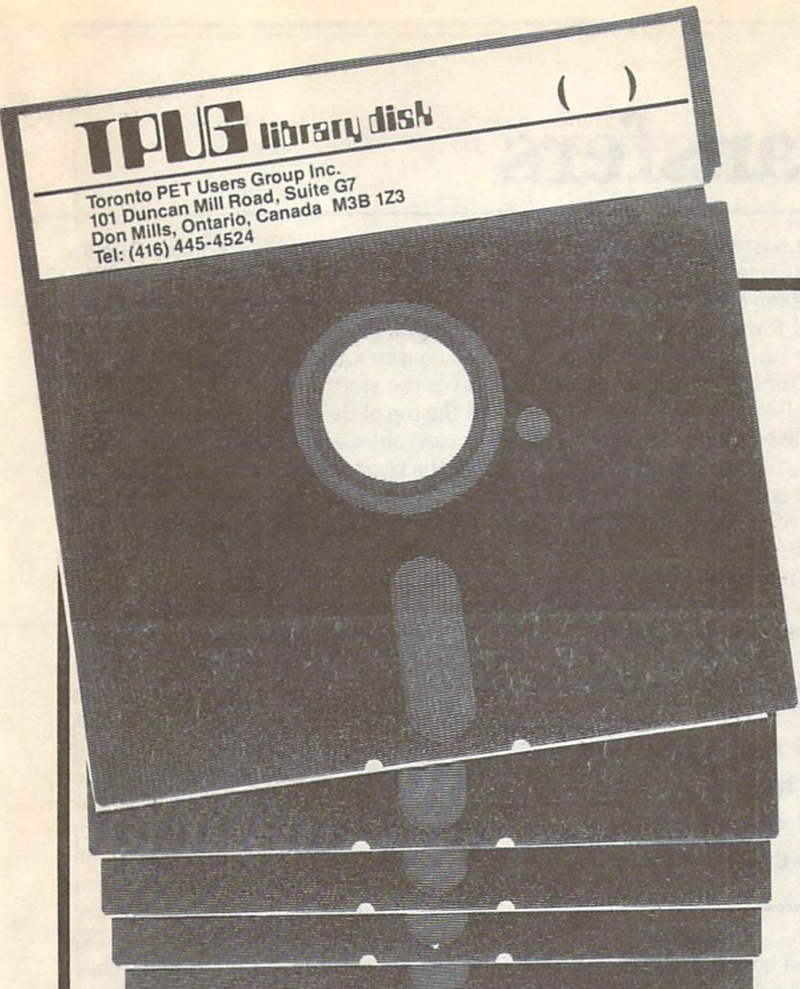
Active PunterNET nodes

PSI-WordPro	Mississauga ON
TPUG	Mississauga ON
East York BBS 1	Toronto ON
USN	Toronto ON
Comspec	Toronto ON
Tech Board	Toronto ON
East York BBS 2	Toronto ON
Amex	Georgetown ON
C-Power	Mississauga ON
Bram-Net	Brampton ON
Bradley 3	Toronto ON
BBBBS West	Mississauga ON
Forest City BBS	London ON
DataCom	Cornwall ON
Q.C.U.G	Belleville ON
Point After BBS	Anaheim CA
Datacom	Highland IN
NWCUBBS	Seattle WA
Commodore/PDX BBS	Portland OR
Keystone C64 BBS	Allentown PA
Prestige Info Net	Alta Loma CA
VCBBS	Vancouver WA
C.U.S.S.H	Racine WI
H.H.N.L	Waukesha WI

VIC Expansion Diagram Correction


CAT No. 276 - 154 A
PIN #1 IS 6 FROM LEFT AND 5 DOWN
HM 6264P
Static RAM
+5V
22 21 17 12 11 10 9 8 7 6 5 4 3 2 1
S R P N M L K J H F E D C B A
VIEW FROM INSIDE THE VIC (LOOKING OUT)

This is the correct version of the component layout and wiring diagram accompanying Ron Byers' article "Expanding your VIC", in Issue #24. The view is from the top, as though the fiberglass board was invisible. Thus you are seeing a mirror image of the etched side of the board (in grey in the diagram). The top-side and bottom-side edge-connector pads are out of alignment in this diagram, but this should have no effect on construction. Be sure to place the IC so that pin #1 is in the sixth hole from the left, fifth hole from the top.
This project was constructed by Assistant Editor Tim Grantham, and it works.


TPUG library disk
( )
Toronto PET Users Group Inc.
101 Duncan Mill Road, Suite G7
Don Mills, Ontario, Canada M3B 1Z3
Tel: (416) 445-4524

TPUG LIBRARY DISKS

10 months of the latest, fabulous public domain software only

$79.95 CANADIAN
US orders US$59.95

Subscribe now to start receiving the TPUG disk of the month

You SAVE 20% or more by using this offer as a TPUG member

THIS OFFER VALID FOR PAID UP TPUG MEMBERS ONLY
PLEASE PRINT

Name: ______________________ Membership # ______________

Address: ____

Type of Payment

☐ Cheque
☐ Mastercard
☐ Visa
☐ Money Order

Amount $ ______________
Currency ☐ Can. ☐ US
Credit Card # ______________
Expiry Date ______________

Signature ______________________________

Type of Computer

☐ C 64
☐ VIC 20
☐ PET ☐ 4040 ☐ 8050
☐ SUPERPET (10 disks) ☐ 4040 ☐ 8050
☐ C128 (1541 only)
☐ MS/DOS
☐ AMIGA (3½") (7 disks) ☐

Tape-to-disk transfers

by M. Garamszeghy

TTD (Tape To Disk) is a simple yet powerful program that will automatically transfer all types of program files, including BASIC and non-relocatable machine language, to disk with the original load addresses intact. All data files are transferred as sequential files.

TTD was originally designed to run on a VIC 20. It can also be run on a C-64 or C-128 (in C-64 mode) by changing the **sys** addresses. Delete line 30 for a C-64 or line 35 for a VIC 20. Although **TTD** will run on any size VIC 20, the maximum length of a program or data file that can be transferred is limited by the amount of RAM available. At least 3k of expansion RAM is recommended for practical applications. Once the program has been entered, it is a good idea to save it before trying to run it for the first time. Since the program uses a number of unconventional entry points into ROM routines, the program could crash or hang-up if an error has been made in entering the ML addresses. In addition, it changes the normal **load** and **save** vectors, so you might not be able to save it after it has been run. It is best to do a hard reset before trying to run any other program.

Prepare yourself by getting out your tapes and a supply of formatted disks, then run the program. After the introductory blurb, you will be instructed to insert a disk into the drive and to press **PLAY** on the tape. At this point, **TTD** takes over and will transfer the entire side of the tape to disk. Unfortunately, there is no elegant way to stop **TTD** once you reach the end of the tape. The simplest way to stop the program is by pressing the **run-stop/restore** key combination. Disk errors which are detected but cannot be corrected by **TTD** are displayed on the screen. If they can be corrected by the user, the program can be resumed by pressing any key after corrective action has been taken. When a disk full error is encountered, for example, the program resumes by resaving the last file loaded.

TTD works by by-passing the normal BASIC and Kernel **load** and **save** routines and substituting custom routines. The **TTD** program resides in the bottom of normal BASIC RAM and is completely relocatable, just like most other BASIC programs. Line 10 hides most of the available RAM from BASIC by resetting the top of RAM pointers. 2K of RAM is used for the actual **TTD** program while the remaining RAM is used as a 'safe' area for the program and data file transfers. Because the entire program or data file being transferred must reside in RAM, the maximum length of a file which can be transferred is limited by the amount of RAM available. In the unexpanded VIC, only 1.5K is available. However, any amount of expansion RAM can be utilized. In the C-64 (or C-128 in C-64 mode), over 35K is available.

. . . TTD can easily be converted to run on many other Commodore computers . . .

To understand how the **TTD load** and **save** routines work, one must have at least a rudimentary knowledge of the structure of VIC 20 tape and disk file formats. In simple terms, a tape file contains two distinct parts: the tape header, which contains data on the file type, filename, and in the case of program files, the starting and ending addresses; and the program or data block. The filename is optional for tape files. The program or data block is located immediately after the tape header. Disk files also have two elementary parts. The filename and file type reside on a special part of the disk called the directory track. The starting RAM address, however, is contained within the first two bytes of the program block. Filenames are mandatory for disk files. The program and data blocks are not located immediately following directory entry, but can be scattered throughout the remainder of the disk.

The loading portion of **TTD** begins with line 40. **Sysa1** calls a BASIC ROM routine which searches the tape and loads the next tape header into the tape buffer (beginning at RAM address 828). Address 828 contains a byte representing the type of file. A value of 1 indicates that the file is a relocatable program; a value of 3 corresponds to a non-relocatable program; while any other value indicates a data file. Addresses 829 and 830 contain the low and high bytes of the start address of the normal load, and 831 and 832 contain the ending address. The rest of the tape buffer contains the filename. Line 40 also saves the start address in a safe location at the top of the tape buffer (900 and 901). These values are needed for later restoring the start address to its proper value. Line 40 is also used to perform a check of the type of file because program files and data files must be handled in a different manner. If a data file is detected by **TTD**, the program will branch to line 200. This will be discussed later.

Lines 50 to 70 calculate the length of the program file and reset the various **load** pointers to the safe area of RAM. The **SYSA2** in line 70 calls the BASIC ROM routine which actually reads the program block on the tape. The subroutine in lines 170 to 190 determines the filename from the tape buffer, calculates its length and prints it on the screen. If a filename is not present (filenames are optional for tape files but mandatory for disk files), **TTD** will assign a single character name based on the value of parameter *c*, beginning with the letter 'a'.

Lines 80 to 100 are used for setting up the various pointers and opening the communication channel required for the modified save operation. The **sysa8** in line 110 jumps into the middle of the BASIC **save** routine to save the rest of the program file. A normal BASIC or Kernel **save** would result in an incorrect load address being sent to the disk because the program was relocated by **TTD**. Lines 120 to 160 perform the disk error handling. If everything is O.K., the program loops back to line 40 to find the next tape header.

Lines 200 to 230 are used to read a tape data file. The first part of line 200 is particularly interesting because it contains a method to allow a data file to be read without having to first open the file with the BASIC or Kernel **open** command. This trick is necessary because, if an **open** statement was used, the internal ROM routines would skip the current file and search the tape for the next data file. The trick is performed by adding the 'new' file characteristics to the tables of **open** file numbers, secondary addresses and device numbers beginning at addresses 601, 611 and 621 respectively, and incrementing the byte at address 152 which represents the number of open files. This has the effect of opening the file without searching for the next header. Line 210 clears the tape buffer of garbage that results from reading the tape header. Lines 220 and 230 get the data and poke it into RAM. Line 240 closes the data file which was never opened!

Although originally written for the VIC 20 and C-64, **TTD** can easily be converted to run on many other Commodore computers that use the 1530 datasette tape format. Line 30 (for the VIC 20) or line 35 (for the C-64) contains the addresses for the Kernel and BASIC ROM routines used by **TTD**. The only conversion required to run on other machines is to change the values of A1, A2, A3, A7 and A8 to the corresponding addresses for the target computer. The following table may be of interest to users wishing to convert **TTD** to run on other machines:

Address	ROM Routine
a1	Find next tape header
a2	Read tape **load** block
a3	Send secondary address (part of BASIC **open**)
a7	Reset tape pointer
a8	Save program bytes (part of BASIC **save**)

Corresponding addresses for various machines can be obtained by consulting either a detailed memory map or a disassembly of the ROM's.

Most programs will be ready to run as soon as they have been transferred to disk. Some programs, however, such as multipart BASIC programs and programs written to access tape data files exclusively, will have to be modified slightly in order to run from disk. First, let's deal with multipart programs. At the end of the first part (and the second part of a three part program), there will be a statement similar to **load"part2"**; or simply **load**. For disk operation, this should be replaced with a statement similar to **load"part2",8**. Most commercial and public domain software can be used with either tape or disk data files without modification. However, some programs which access data files may not be able to use disk files without some minor modification. For tape file access, a BASIC program will contain a line similar to **open1,1,1,"filename"**. For disk files, the statement should be modified to one of the following forms:
open1,8,8,"0:"+"filename"+",s,r" (to read a file); or
open1,8,8,"0:"+"filename"+",s,w" (to write a file). Subsequent **get#**'s, **input#**'s and **print#**'s are the same for both tape and disk files and can be left as is in the original program. □

```
10 print"<clr><rvs>ttd":poke56,peek(44)+8:clr:m=peek(
   56)*256+256:pokem-1,0
20 print"<2 down>prepare disk for save":m1%=m/256:m2%
   =m-m1%*256:open15,8,15:c=65
30 a1=63276:a2=63562:a3=62421:a4=65457:a5=65427:a6=65
   448:a7=64398:a8=63012
35 a1=63276:a2=63562:a3=62421:a4=65457:a5=65427:a6=65
   448:a7=64398:a8=63012
36 rem delete line 35 for vic-20 or line 30 for c-64
40 sysa1:poke900,peek(829):poke901,peek(830):ifpeek(8
   28)=2orpeek(828)>3then200
50 pl=(peek(832)-peek(830))*256+peek(831)-peek(829):m
   h=pl+m:poke832,mh/256
60 poke831,mh-peek(832)*256:poke829,m2%:poke830,m1%:
   poke193,peek(829)
70 poke194,peek(830):gosub280:sysa2:i=848:gosub170
80 poke780,8:poke781,8:poke782,1:sys65466:poke780,l:
   poke781,65:poke782,3
90 sys65469:poke172,peek(829):poke173,peek(830):gosub
   280:poke193,peek(829)
100 poke194,peek(830):poke780,97:poke782,l:sysa3:poke
    780,8:sysa4:poke780,97
110 sysa5:poke782,0:sysa7:fori=900to901:poke780,peek(
    i):sysa6:next:sysa8
120 input#15,e,e$:ife=63thenpoke832+l,c:c=c+1:gosub17
    0:goto80
130 ife=72thengosub160:goto80
140 ifethenprint"<rvs>disk error",e$:end
150 goto40
160 print"<rvs>disk full: insert new disk<down> hit a
    <space>key":poke198,0:wait198,1:return
170 ifpeek(i)=32theni=i-1:ifi>833then170
180 l=i-832:ifl=1thenpoke833,c:c=c+1
190 f$="":fori=833to832+l:f$=f$+chr$(peek(i)):next:
    printf$:return
200 poke602,1:poke612,1:poke622,96:poke152,2:mh=m-1:
    print"<rvs>";:i=848:gosub170
210 fori=828to1020:pokei,0:next
220 get#1,a$:ifa$=""then220
230 mh=mh+1:pokemh,asc(a$):ifst=0then220
240 close1:open8,8,2,"0:"+f$+",s,w":fori=mtomh:print#
    8,chr$(peek(i));:next
250 print#8:close8:input#15,e:ife=72thengosub160:goto
    240
260 ife=63thenf$=left$(f$,len(f$)-1)+chr$(c):c=c+1:
    print"<rvs>";f$:goto240
270 ife=0then40
280 poke175,(mh+1)/256:poke174,mh+1-peek(175)*256:
    return
```

Micro Processes

DIR and LIST Under AmigaDOS

by Betty Clay

Copyright © 1986 Betty Clay

Almost anything seems hard until you learn to do it. Then it is so simple! And so it was with getting a directory of the files on the Amiga drive.

In the first days with our early Amigas, there was no documentation available. By experimentation, we learned to get the current directory by typing **dir** at the prompt in the **cli** window. The directory tends to scroll rapidly, but it can be halted by pressing the space bar, and restarted by pressing **ctrl-x**. When developers' kits began to become available, complete with manuals, we learned to use the options that go with **dir**:

• **dir opt a** lists the files in the current directory and in any subdirectories belonging to the current directory;
• **dir opt d** lists only the directory names from the disk;
• **dir opt i** lists the files and directories one at a time, with a question mark after each name. Proper responses to the question mark are: **return** (to get the next name); **q** (to quit the program); **b** (to go back to the previous directory level); **e** (used when a directory name is displayed to enter that directory and display the files and subdirectories under it. It is possible to use **e** and **b** to toggle between levels); **del** (to delete the file whose name is displayed – if a directory name is displayed, it can be deleted only if no files are contained in it); **t** (to type the contents of a file to the screen; **ctrl-c** exits the **type** mode); and **?** (to display a list of currently valid responses).

If you want the output from **dir** to go to a device other than the screen (a disk file or a printer, for example), you must use the > redirection operator, like this:

dir >prt:

The list provided by **dir** is alphabetically sorted, but does not provide detailed information, such as size and creation date. To get that kind of information, you need the **list** command. **List** provides six fields of information for each file or subdirectory: **filename**, **size**, **protection**, **date**, **time** and **comment**.

• **filename** is the name of a file or directory;
• **size** is the number of bytes in the file;
• **protection** indicates the access status of a file (usually it is 'rwed', which means read, write, execute and delete operations are all allowed);
• **date** and **time** tell when the file was created;
• **:comment** prints a comment made using the **filenote** command, if one was made.

The **list** command offers many options. Some examples:

• **list "fonts" to prt:** prints **directory "fonts" on Monday, 16-dec-85**. Then follows the list of files in the **fonts** directory, with the size, protection, date and time. Notice that you use the **to** keyword with **list**, not the redirection operator, if you want to divert its output to a destination other than the screen.
• **list to prt: fonts keys** gives the same information, but also provides the block number of each file.
• **list fonts keys nodates** displays the list on the screen (actually, the current window), but omits the dates and times of creation of the files.
• **list fonts keys since monday** displays lists only those files in the **fonts** directory that have been created or updated since Monday. **monday** can be replaced with a date in the form **15-Dec-85**.
• **list to df0:oldfonts fonts keys upto 17-dec-85** will list those files in the **fonts** directory that were created or updated on or before the specified date. The list is written to the specified file, here **df0:oldfonts**.
• **list fonts quick** displays only the names of the files and directories. It is much like **dir**, but is not alphabetically sorted, and is displayed in one column, not two.
• **list to prt: fonts s gar** prints only those files whose names contain the string 'gar'.

The syntax for **list** permits the use of either upper-case or lower-case letters in all fields. Commands or names are separated by spaces only. Instead of **prt:** or a disk filename in the commands above, you can use the name of another device if you wish the list to be sent to the serial part (**ser:**), the parallel port (**par:**), and so on. As usual in AmigaDOS, filenames can be in quotes or without quotes, although the quotes are compulsory if the name contains spaces or certain other special characters. □

Automodems: Auto Chaos

by Phil Kemp

My new Telewarbler autodial modem won't work with the WonderTerm program. Does anyone know why?

Messages like this keep appearing on bulletin boards; they are symptoms of a real disease. The disease is called 'lack of standards'; it spreads when independent hardware manufacturers choose independent ways to implement the 'automatic' features of modems.

Advertisements, of course, do not call attention to this issue. *This modem package comes complete with compatible software to fit all your needs.* But computer users are often imaginative souls, well able to dream of a function that the supplied software just *can't* do. And there the fun begins. Often, the supplied software is in machine code: great for efficiency, less great for understandability. Mostly, it comes without documentation as to how it works. In some cases, the program is copy-protected, making it difficult for even knowledgeable users to examine, understand and change it.

There is not just one use of the word 'automatic'. Hayes & Clones, Inc. make 'Smartmodems'. These scan the outgoing characters for sequences that are recognized as commands. We can type in **+ + + ATDT4218765**, then sit back and listen to the burps of the autodialling. On the other hand, Commodore and its immediate imitators sell less expensive automodems. Though physically able to make the right dialling noises, these boxes generally need to be told which noise to make, when to start making it and when to stop. The 'automatic' part of dialling then, is achieved by logic in our terminal program — which is where the 'supplied compatible software' comes in.

We usually connect our modem to the user port of the 64. There is general agreement as to which pin of this connector is used for output bits, which for input data and so on. Unfortunately there is no agreement on which pins to use to control autodial features. Manufacturers have filled this void with diverse schemes. Hence we need a terminal program with logic to suit the particular automodem we buy.

Some programs (the public-domain **Universal Modem**, for example) have options to suit a range of popular automodems. Does it suit the modem I plan to buy? Does the program that comes with the modem have all the features I'm likely to want? These questions demand answer, before I commit my money to a product.

Automodems are desirable and practical items; I plan to acquire one. Manufacturers are to be commended for making them available to us at modest prices. However, be aware of potential problems before you buy, and get all the advice you can.□

Building a $20 Utility Stand

by Edward K. Crossman

None of the commercially-available computer utility stands that I could find were quite suitable for my Commodore equipment. Either they only had one shelf, or the shelf was not deep enough to support a 1541 disk drive correctly. To get a suitable stand I would have had to buy an entire desk/stand arrangement costing 75 dollars or more. But I already had a table — no need to buy another. The stand that I built from necessity, and describe in this article, will not only accommodate Commodore equipment, but many other computers also.

If you are the least bit handy with tools, you can build this stand in a few hours with very little effort and be proud of the result. The two-shelf arrangement is nice because it puts the TV (or monitor) at eye level and locates the disk drives where disks can be inserted or removed without warping them. Also, there is sufficient space both above and around the 1541 drive for heat dissipation (diagram 1). The ten thousand cords that seem to emerge from the rear of my C-64 fit nicely on the table beneath the lower shelf, and are mostly out of sight.

The two end-pieces (diagram 2) can be made from 3/4-inch particle board (not chip board) and are 11 1/8 inches high and 12 inches wide. The shelves are 3/8-inch plywood (AC grade) and measure 18 inches long by 12 inces wide. After cutting the shelves and end-pieces, the next step is to rout out channels in the end-pieces. The shelves will slip into these channels. To make the channels, use a 3/8-inch straight routing blade that will cut

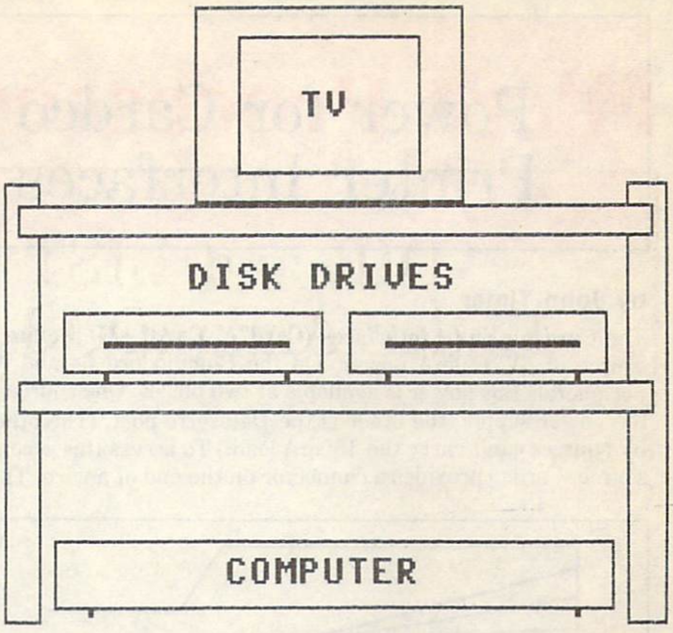


FIG. 1

a groove that is flat on all three sides. The channels should be no more than 1/4-inch deep, and it is probably best to make two passes (1/8-inch cut per pass) rather than a single pass.

Then, using a 1/8-inch drill bit, drill three equally-spaced holes in each slot. Use a countersink bit on the outside of the end-pieces so the heads of the woodscrews will be flush with the outside surface. A #6 flathead screw about 3/4-inch long should be sufficient to draw the surfaces together while the glue sets.

If you don't have a router to cut the shelf channels, you could always glue and nail a support brace (1/2-inch square by 12 inches long) on the inside of each end-piece just below each shelf. Then apply glue to the end of each shelf, which will butt against the end-piece, and to the bottom of the shelf where it rests on the support brace. I would still recommend using the woodscrews, as they will pull all pieces together to insure a tight glue joint.

Now you are ready to assemble the pieces. Place a good grade of wood glue in each slot as well as on the end of each shelf. Then place the shelves in the slots and screw the endpieces to the shelves. Wipe away any excess glue with a damp rag. After the glue dries, sand any rough edges and apply two coats of paint. Voila! You're the proud parent of a two-level stand!□

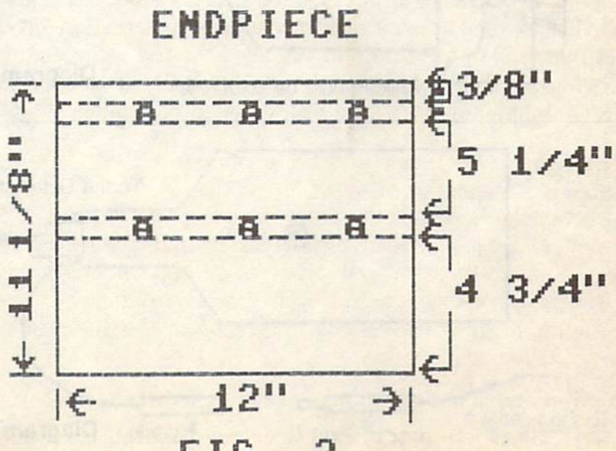


FIG. 2

Power for Cardco Printer Interfaces

by John Timar

The Cardco printer interfaces (**Card?A, Card? + G**) require a source of 5V 150mA power. On the Commodore 64 and its peripherals this power is available at two places. One source is the power supply; the other is the Datasette port. (The other 5V sources can't carry the 150mA load.) To access this second source, Cardco provides a connector on the end of a wire. This connector is plugged in between the Datasette and the computer. Electrically this set-up is fine, but it makes the Datasette plug stick out about 7 cm (nearly 3 inches) – making the Cardco connector susceptible to breakage.

The following arrangement eliminates the need for the connector and, by making the power supply wire longer, allows more freedom in locating the computer, the disk drive and the printer.

- Using a small Phillips (cross-head) screwdriver, open the Datasette plug and, with the wires attached, remove the white plastic terminal from the plug. There are 7 wires going to the 6 pins (pin #1 has two wires attached). See diagram 1.
- Separate the green wire (pin #2) by slipping a strip of cardboard between the green wire and the other wires.
- Strip about 1 cm (3/8 inch) of the green wire. See diagram 2.
- Attach the end of a new wire (length as desired – at least 10 cm, preferably longer) to the stripped section and solder it. See diagram 3.
- Cover the splice with electrical tape and remove the cardboard.
- Reassemble the plug.
- Place a miniature jack on the other end of the wire.
- Remove the connector on the end of the single wire coming from the DIN plug of the interface. Replace it with a plug of the same size as the jack used. (To avoid accidental short circuits, use the inside terminals of the jack and the plug; on many models the outside terminals are exposed.) See diagram 4.

If you want to ensure that you can use your printer without the Datasette, solder another wire to the second terminal of the interface connector (the terminal the power wire was originally attached to) and put another jack on the other end of the wire.☐

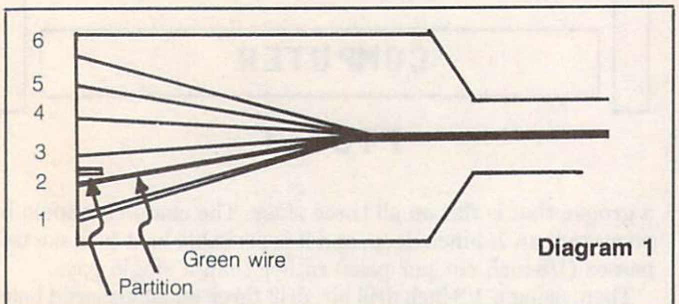


Diagram 1

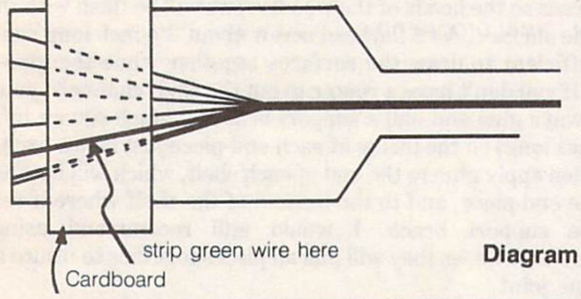


Diagram 2

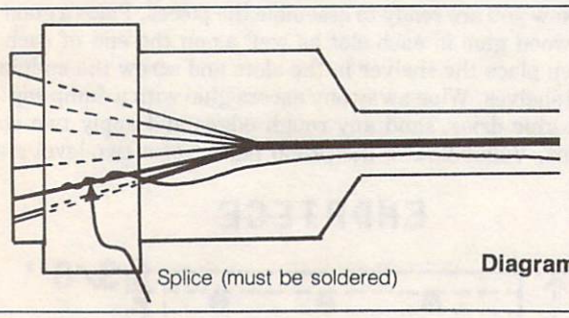


Diagram 3

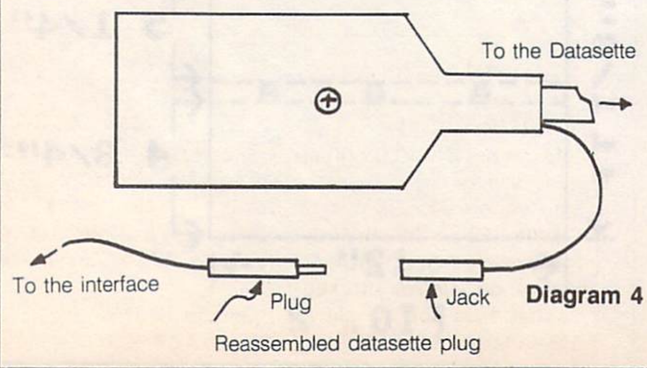


Diagram 4

Prg software
22371B 119 Avenue
Maple Ridge, B.C.
V2X 2Z2

PRGS

MONEY MASTER: available now for the Commodore 128
Fast, flexible, and comprehensive home accounting. Accounts, journal, and budget follow similar methods of operation. Enter, edit, load, save, and print are available for each. Trial balance sheet, and budget variance reports can be printed. Use menus to select from thirty functions. Windowing is used with 80 columns to help keep track of what is going on.
Money Master includes limited mail and telephone support.

SOURCE LINKER: available 15 July for the Commodore 128
The program that makes it possible to reuse parts of your last BASIC program without rewriting it. Transform programs in a disk to disk operation.
Join multiple modules, select parts of a program, compress code, rename variables, renumber lines, print formatted listing, print variable cross reference, list differences between program files, or print line number cross reference.
Source Linker includes limited mail and telephone support.
A complete software purchase comes with a disk program "ready to go", a comprehensive user guide, and an annotated source listing to help you customize the program. You don't want to pay for a source listing if you don't intend to use it. You don't want to pay for a user guide if you would rather just load and go and see how it works. You don't want to pay too much for any of them.

MONEY MASTER Prices:

Program Disk	9.95
User Guide	19.95
Annotated Source Listing	19.95

SOURCE LINKER Prices:

Program Disk	9.95
User Guide	19.95
Annotated Source Listing	19.95

Please include payment in full with order. BC residents must add 7% Provincial sales tax.

AVAILABLE SOON:
- 80 column bit mapped graphics
- database management package
- BASIC libraries including
  - screen manager
  - multi-keyed disk access, and
  - memory table handler.

  Use them with Source Linker.
- plot package with 3D coordinate transformations and multi-device output
- video store inventory management

Expand Past Maximum Capacity!

The Tech/News Journal For Commodore Computers

At better book stores everywhere! Or 6 issues delivered to your door for just $15.00 (Overseas $21 U.S. Air Mail $40 U.S.)
The Transactor. 500 Steeles Ave. Milton, Ontario. L9T 3P7.
416 878-8438
Also check out The Transactor Disk and The Complete Commodore Inner Space Anthology - to us, expansion knows no limits!

"Cecil's 6502/6510 Conditional Macro Assembler" and "Assembly Language Source Code Editor" for Commodore 64® or Commodore 128® or Commodore SX64®

A package of programming tools on one 5¼ **DISK** created for every assembly language programmer.

Fully capable for the professional: "'Cecil's Assembler' always EVALUATES THE WHOLE EXPRESSION, THEN RETURNS THE CORRECT HIGH/LOW byte".

Cecil's Editor provides easy access to extensive on-line help screens for a variety of information about this assembler package. On-line help describes the assembler syntax, pseudo-ops and macro instructions, and the editor commands.

- *Features 10 built-in macro instructions to complete the branch or jump on condition instruction sets.*
- *Labels may be up to 13 characters in length. All 13 characters are considered significant.*
- *Includes full-screen editor with enhanced DOS commands.*
- *Nests macros to 32 deep.*
- *Includes 10 pseudo-ops to control the assembly process.*
- *Stops on error and/or identifies error type and location.*
- *Supports most printers and interfaces and allows multiple disk drives.*
- *Prints its own instruction manual.*

ONLY $49.95 Cdn. or $34.95 U.S. by Check, Money Order, Visa or MasterCard to:

NORTH OHIO FIRMWARE
Box 2661
North Canton, Ohio 44720-0661

( ) English	( ) C64®
( ) German	( ) C128® in C64® Mode
( ) French	( ) SX64
( ) Information $1.71 Cdn.	$1.00 U.S.

Or Call 216/497-0736 from 9:00 a.m. to 11:00 p.m. EST.

Commodore 64®, Commodore 128® and Commodore SX64® are registered trademarks of Commodore Business Machines.

ATTENTION

ALL COMMODORE 64, VIC 20, COMM. 16 AND COMMODORE 128 OWNERS

A complete self-tutoring BASIC programming course is now available. This course starts with turning your computer on, to programming just about anything you want! This course is currently used in both High School and Adult Evening Education classes and has also formed the basis of teacher literacy programs. Written by a teacher, who after having taught the course several times, has put together one of the finest programming courses available today. This complete 13 lesson course of over 220 pages is now available for the COMMODORE 64, VIC 20, COMMODORE 16 and the COMMODORE 128and takes youstep by step thru a discovery approach to programming and you can do it all in your leisure time! The lessons are filled with examples and easy to understand explanations as well as many programs for you to make up. At the end of each lesson is a test of the information presented. Furthermore, ALL answers are supplied to all the questions and programs, including the answers to the tests. Follow this course step by step, lesson by lesson, and turn yourself into a real programmer! You won't be disappointed!

We will send this COMPLETE course to you at once for just $19.95 plus $3.00 for shipping and handling (U.S. residents, please pay in U.S. funds). If you are not COMPLETELY satisfied, then simply return the course within 10 days of receipt for a FULL refund.

FOLLOW UP COURSE

Now available! a 200 page course exclusively on sequential and relative files using a unique approach for those with very limited file programming experience - set up your own personal and business records! — disk drive a must — same author — same guarantee — same cost— this course for all computers except Vic 20.

Fill in the coupon or send a facsimile.

TP

NAME:________________________

ADDRESS:________________________

CITY:________________________

PROV./STATE:________________________

POSTAL/ZIP CODE:________________________

I desire the BASIC programming course for:
Commodore 64 ☐ Vic 20 ☐
Comm. 128 ☐ Comm. 16 ☐
I desire the FOLLOW-UP ☐ course on relative and sequential files for all above computers but Vic 20).

Any complete course:	**$19.95**
Postage and Handling:	**$3.00**
Total:	**$22.95**

Send Cheque or Money Order to:
Brantford Educational Services
6 Pioneer Place,
Brantford, Ontario,
Canada N3R 7G7

Library Additions

This month we present our first MS-DOS disk, for those of you with PC10s or PC10-compatible machines such as the IBM PC, and we welcome TPUG's MS-DOS librarian, Colin Justason, to the pages of the magazine. We'd also like to bring to your attention a change in our disk-naming convention to four letters from the old three — we had at last run out of letters under the old system. We hope you find it less confusing than we do.

CP/M disks (Z)AAA and (Z)AAB

Presented by Adam Herst

Many of the programs in the CP/M library are contained in **.lbr** (library) files. To remove them from the library use **lu.com** (library utility) contained on TPUG disks (Z)AB and (Z)AC. Type **lu** to start the program; **-o** opens a file. Then type the library file name. Typing **-l** will give you a listing of the files contained in the library, while **-e** followed by the filename to extract a particular file from the library. Wildcards (? and *) can be used to extract multiple files. See the accompanying documentation for more information.

Some of the extracted files will be in a squeezed format. This is indicated by a filetype of **.xqx** where the **q** indicates a squeezed file. Use **usq.com** or **nswp.com** to unsqueeze these files.

Disk (Z)AAA (no relation to disk (Z)AA) is a languages disk. Some of them are better documented than others, and none has been exhaustively tested on the C-128. **smalc1.lbr** is a version of the **Small-C** C compiler. This is a large library of many small and a few not so small files. Most of the files have accompanying documentation. Only the **.com** files are included. While the source files are available, no one has made them available to me. The documentation files that are included detail the specifics of this compiler; they are not a tutorial in C.

lllbasic.lbr contains all the relevant files for this BASIC interpreter: source code, runtime interpreter and documentation file.

Unlike the previous file, this implementation of the FORTH language, **forth.com**, comes with nothing but the **.com** file. If you can figure it out and put together a documentation file, please send it in.

```
zaaatype me    forth123 com
lllbasic lbr   smalc1   lbr
```

Disk (Z)AAB is primarily a disk of utilities. For the assembly language programmer we have **Cpm3lib.lbr**, which contains a number of CP/M Plus ML subroutines. Documentation is included. **Z80asm.lbr**, as the name suggests, is a Z80 assembler package. **Xref.lbr** is an ML cross-reference utility program. Documentation is included. **Neat.lbr** contains programs to tidy up assembly language program listings. Documentation is included.

In the way of disk utilities we have **dir+.com**, a file handling program. Enter a ? for the command menu. **Find.com** searches disk/files for a specified string. Enter **find** with no arguments for usage instructions. **Unerase.com** searches and recovers *lost* files. **Xdir.com** produces an extended directory listing.

```
zaabtype me    backga   com
calc     lbr   cpm3lib  lbr
dir+     com   find     com
listt    com   neat     lbr
sargon   com   unerase  com
vde      com   xdir     com
xref     lbr   z80asm   lbr
```

In the grab-bag this month we find **listt.com**, which produces formatted output to the screen or printer. Enter **listt** with no arguments for usage instructions. **Vde.com** is a very good text editor. Enter **ESC ?** for the instruction menu. **Calc.lbr** simulates an HP calculator while exhibiting the internal logic on the screen. Documentation is included. The **term** files let you customize installation.

To finish things up we have a couple of games. **Backgam.com** is a version of the popular board game. **Sargon.com is** a version of chess. That's it for this month, folks — more than enough to keep you busy. Over the summer, I will continue to release CP/M disks as the programs trickle in. Keep up the submissions and have a great summer!

VIC 20 disks (V)AD and (V)AE

Presented by Richard Best

Once again we are giving you a little more. Disk (V)AD will be the last monthly disk to be issued before the summer. It features some very useful stuff plus an excellent Telidon-style graphics demo. We also have **pods**, a statistics package prepared by CP/M librarian Adam Herst.

Disk (V)AD starts out with three peculiar demos called **brain bender** numbers 1 through 3. I won't spoil your fun by telling you what they do.

The school year is drawing to a close — the perfect time for **semester calc**. Teachers with a lot of marks to work into final grades will appreciate this one. It inputs any number of tests, labs and assignments and produces a final mark. You can then use **statistics** to analyse

```
0     vic disk (v)ad
8     "list-me (v)-ad/1"  p
6     "list-me (v)-ad/2"  p
2     "brain bender1.v"   p
2     "brain bender2.v"   p
3     "brain bender3.v"   p
7     "semester calc.v"   p
10    "statistics.v"      p
3     "binary fract.v"    p
6     "rs232 sound.v"     p
2     "screen print.v"    p
1     "vic scrn copy.v"   p
28    "bay street.v12k"   p
9     "500 file.v12k"     p
3     "pix load ii.vsx"   p
13    "no-ghost"          p
3     "no-ghost+"         p
13    "amer. flag"        p
3     "amer. flag+"       p
13    "sr-71"             p
3     "sr-71+"            p
13    "arrow"             p
3     "arrow+"            p
13    "can. flag"         p
3     "can. flag+"        p
13    "magnum"            p
3     "magnum+"           p
13    "wolf ii"           p
3     "wolf ii+"          p
13    "stone"             p
3     "stone+"            p
6     "dir"               p
```

how the class has done or derive other common statistics from a variety of data.

For you hackers, we have something very interesting. **Binary fract** is a conversion program that works on binary fractions. A unique program is **rs232 sound**, which turns a bit of Bach over to the RS232 port. This approach lets you play music in the background while something else is going on in the foreground.

Screen print is a printing utility that will send screen text to the printer. **Vic scrn copy** is a screen dump utility that can be added to your programs as a subroutine. **500 file** is a tape-oriented, single-item file manager. It's menu driven and very easy to use. It will input, sort, print and file up to 500 items.

Bay street is a stock market game/simulation. Up to six investors can play along with the VIC, which acts as the banker. As stocks fluctuate, their values are posted on a chart.

SuperExpander fans will enjoy the demos that make up the remainder of the disk. They are accessed through a program called **pix loader**, which prints a menu and loads the eight picture files from disk. This graphics display is impressive.

Disk (V)AE contains the program **pods**, which stands for Printer-Oriented Descriptive Statistics. It is a statistics package that will run on an expanded VIC with either tape or disk. The disk/tape contains two versions of the program. One is in ordinary BASIC, and the other is *scrunched* to run faster. Also included is five pages of documentation in a sequential file, which you can print using your favourite sequential file reader.

This statistics package is complex. It can analyse data at various levels. You can also graph your analyses and print them on a printer. The whole thing is menu driven and will work with files on either disk or tape. For the user with advanced statistical needs, this could be a very useful program.

The error that was present in **p-term22** has been repaired and the library disk/tape has been remastered. If you already have this issue and want to use the terminal program, see *TPUG Magazine* issue #23 for the correction. This program does indeed work and seems to do all the things it says it does.

Finally, to all you VIC fans, have a great summer, and do keep writing for the VIC. Your contributions to the TPUG library are appreciated all over the world. (Besides, we'll send you a disk/tape of your choice in return for a submission.)

MS-DOS disk (M)AAA

Presented by Colin Justason

This is the first of the TPUG PC disks, and I have tried to make it a bit of an all-around disk, with something for everyone. It is, for the most part, a shareware/freeware disk, and I suggest that if you are going to use these programs you should pay the programmer something for his effort if you find the program useful. Most of the programmers make contribution suggestions that do not seem out of line — in fact most are a steal for the quality of their work!

```
msdos (m)aaa
read      me
printme   bat
-fm-ffm-  ---
fm        exe
ffm       exe
-nu-epsn  ---
nu-epson  com
nu-epson  doc
-pops---  ---
popsetup  com
popsicle  com
-pmap---  ---
pmap      doc
pmap      exe
-ddir---  ---
ddir      com
-rendir-  ---
rendir    com
-wrp----  ---
wpk       doc
wpk       exe
practice  wpk
rhymes    wpk
story     wpk
thecat    wpk
thedog    wpk
-clock--  ---
clock     com
-tired--  ---
tired     com
-pc-tch-  ---
pc-touch  doc
pc-touch  bas
pc-touch  exe
pc-touch  fil
```

Back to the disk. The first two programs on the disk are **fm** and **ffm**, and can display two directories or disks at a time, great for comparison or copy work. Instructions are available at any time by using the function key **F2** for either program. **Fm** is very good for monochrome, while **ffm** works best on a colour or hi-res monitor. It includes a show and execution system for running files without leaving **ffm**.

The next program is for Epson or Epson-compatible users who like being able to control their printers from the keyboard. Please read the **nu-epson** docs for special features; however, the menu selection is quite good for a resident program. **Popsicle** is a resident program that is called up whenever you want via your selection of keys which, along with other options, you assign first by running **popsetup**.

Pmap is a neat little program for finding out how much memory all these little memory resident programs take up and where they are in memory. Read the docs for additional features. **Ddir** shows a dual directory and provides an alternative to the **dir/w** or **dir/p** method of finding out what is on your disk. Use it just like you would **dir**. **Rendir** stands for *rename directory* and is for those of you who are into subdirectory uses. Remember to include the path name when you go to rename.

Wpk stands for *word processor for kids*, and it's quite good. Just load and enjoy. There are five examples as well as the docs so no one should have any problems. **Clock** is a very small memory resident program that puts a clock up in the top right hand side of your screen. **Tired** is a little program you can leave on your machine for snoops. **Pc-touch** is a typing tutor for those of us who never have become good at the keyboard (or is my 18 words a minute okay?). Read the docs and give it a try, you may improve your speed. That's it for disk (M)AAA; I hope you enjoy it and remember the programmers.

TPUG BBS

The NEW telephone number is:

(416) 273-6300

Operating hours:
24 hours per day
7 days per week
The password is...

APPLE

C-64 disks (C)HF, (C)TR and (C)AA

Presented by Paul Kreppenhofer

These are C-64 disks that have been released since January.

Disk (C)HF contains an assortment of pictures drawn using a KoalaPad. These pictures can be accessed by loading **run-me**.

Disk (C)TR, our February monthly disk, contains two Ontario income tax programs: **tax85ont v0.1** is a printable, standard tax program, while **taxcalc 85** is for more advanced tax applications. For utilities this month, there are four programs. The first is called **compiler4.4.c**, which will convert your basic programs into faster-running machine language. Next we have **1541 saver.c**, which helps stop drive rattle. Third is a program called **calender.c** that will print calendars in French and English. Finally, there is **mortgage sarnia**, a program that will calculate mortgage payments, including lawyers' fees.

Comma sense.c and **patterns.c** are two educational programs. The former is designed to teach you about commas, while the latter will teach you about the stars. Other files on the disk include **cards**, a game based on the *Mille Bornes* card game; **lazy letters.c**, which will change your computer's character set; **waveforms.c**, which draws waveforms with harmonics; and finally, **proverbial.c**, which will give you a few laughs with funny sayings and last words.

The main program of the March disk, (C)AA, is a large terminal program called **dark term**. This program contains many options and is very good. **Dark term** is loaded by **dt2-boot**. By way of games this month we have **q&a 64.c** which is a trivia-based game. Next there is **mindbusters.c** which, as the name implies, is a mindbuster, requiring a joystick to move patterns around to match other patterns. The last game is a shoot'em-up program called **starscanner.c**, where you shoot the alien spaceships. A revision of the tax program **taxcalc85** is also on the disk, with **tax85ont V0.1** added for your convience. The last program is **org.chemistry.c**, with which you can create new characters for chemistry.

Amiga disk (A)AAA

Presented by Mike Donegan

This disk was put together by Fred Fish of California: see **readme.dist**. It is similar to disks Amigalib1 and Phase4 #13.

None of the programs on this disk can be run from the **Workbench**: they must be run from the **cli**. Be sure to specify the correct path: for example, if you have the disk in drive 1, you must enter **df1:hello/hello** to load and run the **hello** program. The C source code for each program is also included: these files have a **.c** appended to the program filename. You cannot run these directly.

Amigademo is a graphical benchmark for comparing Amigas. **Amigaterm** is a terminal emulation program with Xmodem upload and download. **Balls** is a simulation of the 'kinetic thingy' with balls on strings where only the end balls move (quick, can *you* come up with a better description?). Anyway, it's cute. **Colorful** shows off the use of hold-and-modify mode. **Dhrystone** is the well-known benchmark program. **Dotty** is the source code for the 'dotty window' demo on the Workbench disk. **Freedraw** is a small 'paint' type program: free drawing, boxes, filled boxes, et cetera. You can have fun exploring gadgets with **gad**. Watch your machine's memory usage with **gfxmem**. Cute and useful.

Halfbrite is a sample program that demonstrates the "Extra-Half-Brite" mode available on Amigas with the new 'Daphne' chip that allows 64 colors in low-res mode, rather than 32. **Hello** demonstrates the creation of a simple window. **Latffp** shows how to access the Motorola Fast Floating Point library from Lattice C. It also demonstrates the tremendous speedup obtained. **Palette** is a sample program for designing colour palettes. **Trackdisk** demonstrates the use of the trackdisk driver — a useful example of 'raw' disk read/write. **Requesters** is a sample program with documentation for building and using requesters. **Speechtoy** — you have to see this one. Be sure to click the gadget that pops up the face. **Speech** is a stripped down version of **speechtoy**. □

```
                         amiga disk (a)aaa

hello(dir)                 speech.c              latffp(dir)
     hello             halfbrite(dir)                 latffp
     hello.c                halfbrite                 latffp.c
balls(dir)                  halfbrite.c           gad(dir)
     balls             palette(dir)                   gad
     balls.c                palette                   gad.c
     bres.c                 palette.c             gfxmem(dir)
     Makefile          amigaterm(dir)                 gfxmem
     poster                 amigaterm                 gfxmem.c
trackdisk(dir)              amigaterm.c           amigademo(dir)
     trackdisk              poster                    amigademo
     trackdisk.c       colorful(dir)                  amigademo.c
dhrystone(dir)              colorful              requesters(dir)
     dry.c                  colorful.c                menu
     drynr                  colorful.doc              menu.c
     dryr                   colorful.info             menu.doc
dotty(dir)             speechtoy(dir)             .info
     dotty                  speechtoy             dirfile
     dotty.c                speechtoy.c           readme.dist
speech(dir)            freedraw(dir)              readme.list1
     poster                 freedraw              tpug-type-me
     speech                 freedraw.c            tpug-type-me.info
```

Software order form

NAME ______

STREET ADDRESS ______

CITY/TOWN/P.O. ______

PROV/STATE ______ POSTAL/ZIP CODE ______

TELEPHONE ______ MEMBERSHIP NO. ______

TORONTO PET USERS GROUP, 101 Duncan Mill Road, Suite G7, Don Mills, Ontario M3B 1Z3 416-445-4524

disks

To order club disks by mail, send $10.00 for each 4040/2031/1540/1541 disk (4040 format), discount price 5-10 $9.00 each, 11 or more $8.00 each; and $12.00 for each 8050/8250 disk (8050 format). We do honour purchase orders from school boards.

These disks are for use with a ______ computer and a ______ disk drive.
Please send me the following:

3 Letter/No. Code	Description	4040 or 8050 Format	Price

Total $ ______ .00

tapes

To order VIC 20 or Commodore 64 library tapes, send $6.00 for each tape.
To order PET/CBM or Commodore Educational Software, send $10.00 for each tape.

These tapes are for use with a ______ computer and a datasette.
If for a PET computer, what model - ______ - BASIC - 1.0( ); 2.0( ); 4.0( )?

3 Letter/No. Code	Description	Price

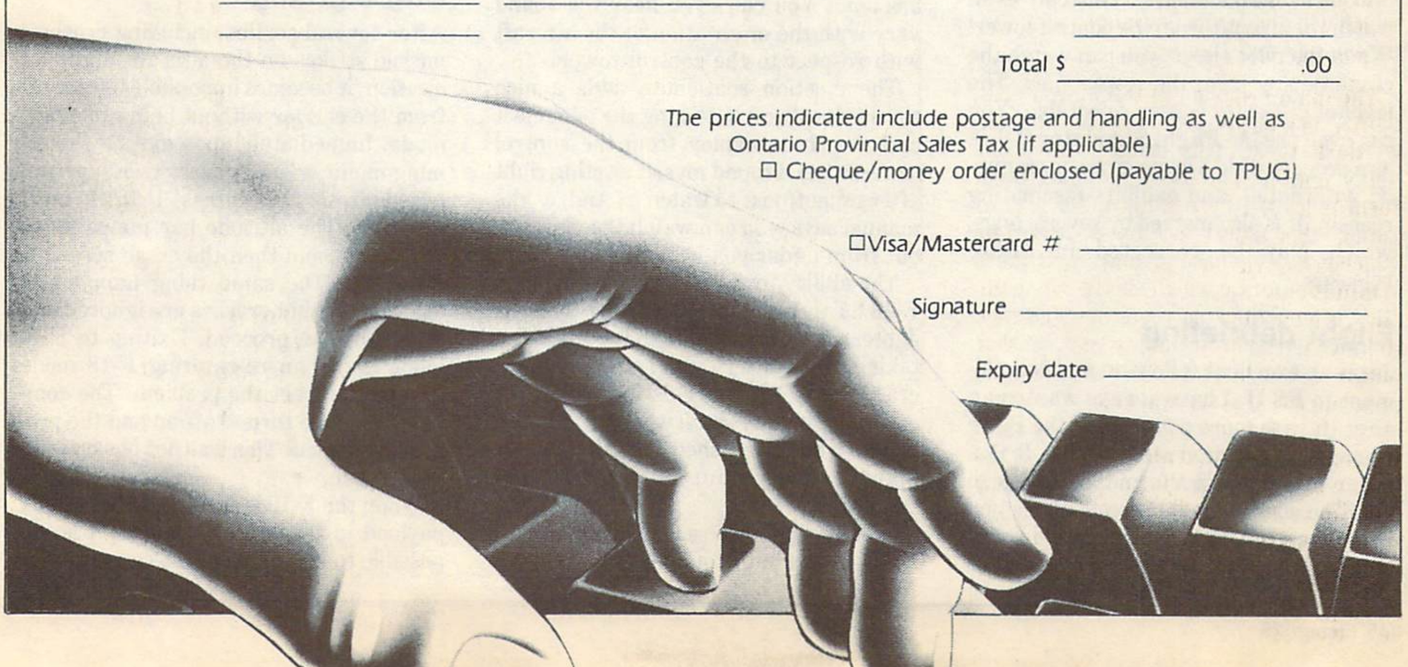
Total $ ______ .00

The prices indicated include postage and handling as well as Ontario Provincial Sales Tax (if applicable)
☐ Cheque/money order enclosed (payable to TPUG)

☐Visa/Mastercard # ______

Signature ______

Expiry date ______



Reviews

Jet
from subLOGIC
Flight simulator
for Commodore 64

Review by Hank Aviles

Jet represents the state-of-the-art in flight simulation for home computers. SubLOGIC has carefully designed this program to work around the visibility limitations of previous flight simulators with a heads-up display that uses over 70 per cent of the available screen area. Other simulators devote roughly 50 per cent of the display to the pilot view, with the remainder used for instruments, weapons status, and so on.

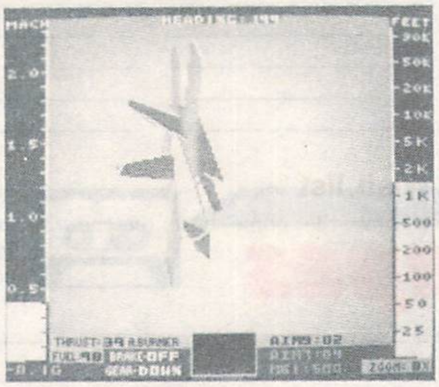
Aircraft speed is indicated by a vertical bar on the left of the pilot display. On the right, another bar indicates altitude on a logarithmic scale. Throttle, fuel, landing gear, speedbrake and weapons status are contained in a narrow horizontal screen at the bottom of the display. Radar, range indicator, and attitude indicator are individually selectable and are superimposed on the pilot's view. The radar is a small square that pops up at the bottom of the display. An innovative feature is the ability to 'zoom' the pilot's view (like a telescope) over a 1 to 8 range.

You can choose to fly a land-based F-16 or carrier-based F-18, in either dogfight or target strike mode. You can even watch the aircraft from the control tower! When the pilot ejects, you can watch the chute deploy from the tower view. The level of difficulty is user-selectable. You can even use the **Flight Simulator II** disk as a scenery disk! The program is crammed with detail, and exhibits fascinating realism. It is also marred by several bugs, which may be corrected in future releases.

Flight debriefing

Jet is an excellent follow-on and complement to **FS II**. I have always wanted an aircraft with more power than the Piper Cherokee simulated with **FS II**. It is a sheer delight to perform acrobatics in **Jet**. The dogfight and target strike action is more convincing than the WW1 game in **FS II**. For those users who want to learn flight physics and light plane operation, **FS II** is still the best choice. For those users that want to have more fun, more power, more action, **Jet** is the choice. The ability to use **FS II** as a scenery disk is a real plus. I enjoyed buzzing the Statue of Liberty and the Space Needle in an F-16 and F-18 jet fighter!

The best feature of the program is the display design. The full-height, square screen display, coupled with the ability to zoom from 1 to 8 times the view area, and front, side, rear and top views give the user superb visibility. The flight instruments are very well done, arranged around the screen to maintain maximum visibility. The ability to select radar, range, and attitude indicators in any combination is great.



The control tower view is fascinating — almost a second game in itself. It was fun trying to fly the aircraft by remote control. It gives you the feeling that you are flying a radio-controlled model airplane. You can even hear the sound vary with the orientation of the aircraft with respect to the control tower.

The ejection seat/chute adds a nice touch of realism. Watching the pilot eject and the chute deploy from the control tower is fun. I found myself ejecting right after takeoff just to watch it. And, as the manual says, you can watch the plane fly out from under you in the cockpit view.

The ability to fly either an F-16 or an F-18 is a great feature. The F-16 is lighter, and more agile than the F-18. But taking off (and trying to land) on an aircraft carrier with an F-18 is a thrill. The carrier looks very *small* when you are trying to land. (I crashed into the carrier many times, but never landed successfully.)

I really missed the zoom feature of the **FS II** radar, but I found that I could get a good imitation by flying inverted at high altitude, and using a low zoom factor.

Jet will not load properly with Epyx's **Fast Load** cartridge even though **FS II** is compatible with this popular accessory for the Commodore 64. With **Jet** I have to disable **Fast Load**, type in **load "0:*",8,1**, and wait three minutes. With **Fast Load**, I can get to the first menu of **FS II** in 60 seconds! A two-key sequence will automatically load and run the program, and subsequent disk accesses are much faster. Both products are heavily copy-protected and rattle the 1541 drive, with or without **Fast Load**.

On **Jet**, the engine has a low throbbing sound that drowns out the higher-pitched turbine tone. With headphones, I was barely able to hear the turbine sound change as the throttle setting was varied. From the speaker on a monitor, the turbine sound is totally drowned out by the throbbing tone. With the afterburner on, it is even worse. The weapons make no sound when they are fired. Even the tone that warns of an enemy missile or plane is very faint.

When in dogfight mode, ground targets cannot be engaged, and vice-versa. It would be much nicer to be able to have dual role sorties, as in the WW1 game in **FS II**.

At times, especially in free-flight mode, both aircraft seemed to become sluggish with extended use. This may be a user perception and/or due to the limitations of the Commodore 64.

Bug sightings

After several sorties, including crashes, missile strikes on the aircraft, and pilot ejection, it becomes impossible to take off from the carrier without being in 'easy' mode. Immediately upon exiting the arming menu, a 'stall' message is superimposed over the 'press 1 to launch' message. The altitude bar jumps up to 25-50 feet, and then the crash screen is displayed. The same thing happens in 'easy' mode, but crashes are ignored and the flight can proceed. Exiting to F-16 mode and then re-entering F-18 mode does not correct the problem. The computer must be turned off/on and the program reloaded. This was not observed in F-16 mode.

When the F-18 is fitted with maximum payload in target strike mode, it is impossible to launch from the carrier if the range circle, and/or attitude indicator, and/or radar features are active. The aircraft does not appear to gain enough momentum before going off the edge of the carrier. It appears to fall off the edge into the water. When a launch is attempted without any of these features active, a successful takeoff can be made. It is tricky, and the rotation has to be timed well, but it can be done. The algorithm that computes the forces acting on the plane may be affected somehow by the reduced simulation throughput with the range/attitude/radar features active. Again, this was not observed in F-16 mode.

During F-18 or F-16 target strike mode, after all the AGM (Air To Ground) missiles have been launched at a target, the MK82 bombs will not release after they are selected. The cannon appears to operate in all cases. Sometimes the bombs will release after you cycle through the weapon choices once or twice. This problem does not occur if the AGM missiles are not launched all at once.

When F-18 free flight mode is selected, the free flight area appears to be as described in the manual, except that the colour of the water the aircraft carrier is sitting in is green, not blue as in dogfight or target strike modes. The green is the land colour used in the F-16 modes. Otherwise, this mode appeared to work properly.

Another bug appeared when using **FS II** as a scenery disk. When a return to dogfight mode was selected, the F-18 was placed on an expanse of blue. After takeoff, flying over the terrain revealed that it was the WW1 game field. The **Jet** program did not detect that the **FS II** disk was present, and did not prompt the player to re-insert the **Jet** disk. The **FS II** WW1 field had black features on a blue background. When I tried this in F-16 mode, I got the same effect, except that the WW1 field had black features on a green background. I was able to arm the aircraft and fire at the background.

In either F-16 or F-18 mode, stall breaks are strange — in fact, non-existent. It is possible to get the aircraft to slide backwards along its flight path without triggering a stall. I did this by getting into a 40 to 50 degree climb, then cutting the throttle to zero. The airspeed decreases to zero, then increases again as the aircraft starts sliding backward. I confirmed this by using the tower view feature. No stall warning was observed. It appeared easier to get the plane to slide backwards than to get a stall. I did get consistent stalls when I tried to climb above 100,000 feet. After the plane stalled at around 100,000 feet, it seemed to slide down along the flight path and start spinning. I was able to recover from the spin after the plane got down to about 50,000 feet.

After getting shot down in F-18 target strike mode, the tower view is sometimes messed up on subsequent flights. The crash/ejection scene from the previous flight is the only thing visible from the carrier control tower. Even a plane on the catapult cannot be observed. The view from the aircraft still functions correctly. Sometimes, when the zoom factor is changed in the tower view, a plan view of the target strike area is superimposed on the old sky view. This usually happens when you try to zoom the view from the tower. I had to reload the program to fix this one. This effect was not observed in F-16 mode.

Perhaps this cannot be called a bug, but the engine sound is not cut after a crash. Actually, about all the sound tells you is whether or not the afterburner is on. This is the most annoying feature of the program.

As a last word on bugs, the word 'missile' is misspelled in the armament menu.

Wish list

- Show the home base on the radar screen. This will make it easier to get back to base.
- Put scale marks on the maps in the manual.
- Have the aircraft's flight performance degrade gradually in combat rather than abruptly, perhaps being able to withstand one or two hits before having to eject.
- Add the ability to follow enemy aircraft and attack from behind as the enemy tries to out-manoeuvre you. Right now, enemy planes approach essentially head-on then circle the F-16/18 in a decreasing spiral.
- Include more specifications on the F-16 and F-18, like performance envelopes, distinguishing features, anecdotes, and so on.
- Show the landing gear when it is lowered.
- Create a level that is between the 0 and 1 levels currently in the program. Right now, the 0 level is very easy, but it is difficult to hit targets and survive at the 1 level on a consistent basis.
- Allow the **Jet** disk to be used as a scenery disk by **FS II**.
- No flight editor is provided with **Jet**, unlike **FS II**. It would be nice to at least have the ability to reposition the aircraft in 3-D space, and change day/night settings.

On a scale of one to five, this program is a *must have* 4.5.

***Jet**, $52.95 (Cdn.), from subLOGIC Corporation, 713 Edgebrook Drive, Champaign, Illinois 61820.* □

40 Great Flight Simulator Adventures and 40 More Great Flight Simulator Adventures

both by Charles Gulick
from COMPUTE! Books

Review by Jim Butterfield

Copyright © 1986 Jim Butterfield. Permission to reprint is hereby granted, provided this notice is included in the reprinted material.

Each of these two books outlines forty flights or challenges to be tried on **Flight Simulator II**. The first book is pleasant, and users may find that it solves the 'what should I do today' problem. The second book (*40 More...*) has a few problems, and doesn't live up to the standard set by the first.

Many owners of **Flight Simulator II** can think up their own flight itineraries. Indeed, documentation accompanying the program gives a number of training flights, and a user can think up many other things to do. Flights from New York to Boston, Chicago to Champaign, Olympia to Seattle and many others can be mapped out using the navigation charts supplied. But if inspiration doesn't come, these books will offer worthwhile suggestions. Instead of drifting around the Chicago area, you can now select one of the flight itineraries and try your hand at it.

In book 1, the flights are graded in order of difficulty. The first few flights are just floating around and sightseeing, and the last few are difficult flying challenges. In between are tutorials and curiosities. A couple of the flights don't work out on the Commodore 64 version of **Flight Simulator II**; flight 26, *Pyramid Power*, demonstrates a curious effect found on IBM versions; and flights 33 and 40 call for implementing turbulence, a feature not present on the 64.

For each flight, the user is given the flying parameters to be entered into the editor. When the editor is exited, you'll find yourself (after an appropriate disk action) at a given location, pointed in a particular direction, ready to pick up the flight from that point.

It seems to me that the books have three types of flights: sightseeing, skill development, and oddities. They are not mutually exclusive: a particular exercise may be a combination of types.

The sightseeing aspect is nicely done. You're given a feel for the area, and details of what's around you. In some cases, features are pointed out that cannot be seen on **FS II** – you're told that it's down there anyway. The author has a nice feel for the countryside, and the descriptions are quite good (although the Statue of Liberty is *not* on Ellis Island as stated).

Skill development is reasonably well graded in book 1. Sometimes you're given a flight itinerary — start here, go to there, perhaps land at a selected location. Other times, you're given a challenge: land with a 'dead stick' (motor off), fly strictly on instruments through fog and cloud, or come down within a very restricted area. And there's just a smidgin of aerobatics in the first book.

Book two starts off nicely as a sequel: the first seven adventures contain 'advanced' flight instruction. The Puget Sound scenery area chosen is more colourful than the Champaign Willard airport suggestion in the **FS II** documentation. Considerable attention is given to the question of patterns: approach and landing patterns are worked through in some detail. The 'seven' adventures are really no more than two or three flights with each flight broken down into phases.

Unfortunately, book two seems to quickly lose its sense of direction. Instead of flight itineraries, the user is given numerous 'tie-down' points: locations to park your aircraft when you're not flying, with vague suggestions about where you might fly from these points if you were so inclined. It's not clear if the author is trying to write 'literature' instead of a user handbook as part of his second volume: for example, adventure 20 is entirely taken up with descriptions of how his budgie, with partly-clipped wings, has made numerous unsuccessful attempts to fly. At the end of the chapter, a paragraph notes that the highest airport in **FS II** is Santa Catalina.

Even worse: adventure 13 of the second volume suggests that you fly due north from Arlington, Washington for two and a half hours. It doesn't say how to avoid running out of fuel (the easiest way is to press the **s** key) and, since there's no scenery out there at all, the author apparently tries to liven up a deadly dull trip with idle chatter (the chapter runs to ten long pages). There's nothing to see on the flight; although the flight takes you a little over 300 miles north of the USA/Canada border, no scenery is available in **FS II**, no navigational aids, nothing to relieve the tedium. Finally, you're allowed to let the plane down, at which point the book announces that you're at the North Pole (within plus or minus 187 feet)! This information will come as quite a surprise to the residents of Prince George, B.C., since that's approximately where you really are.

Oddities are for some people one of the joys of computer programs, and **FS II** has lots of them. The books look at many strange things such as the peculiar ground formation near Sammamish in Washington state, and the way Mount Rainier disappears when you fly to the wrong side of it. Or: what happens if clouds are *lower* than your landing strip? Try volume 1, adventure 24 and see.

Some of the exercises may be too tough for a C-64 flyer (the C-64 plane is hard to fly). And the C-64 aviator should know a few extra facts not covered by the book. For one thing, if you crash, you *won't* return to the start of the adventure unless you have saved it as a 'resident mode', or unless you press the **s** key as the adventure starts. The author found that the reliability factor had little effect. . . if you set 'reality mode' on the 64, this factor will become noticeable, fast.

The books throw in quite a bit of 'flying talk'. For those not in the club, it would be useful to have a glossary defining such terms as 'flying the box', 'classic approach configuration', and 'slow flight'. The book also assumes you know how to handle the controls and navigation instruments — information you can get from the **Flight Simulator** manuals.

I wish the books came with an index. That way, if I wanted details on, say, Santa Catalina or Boston airports, I'd be able to turn to the chapter where they are discussed. In the same vein, I wish that the 40 adventures carried the 'adventure number' as part of its chapter heading: it would make lookup quicker than using page numbers. Book two, by the way, includes a copy of the navigation charts that are included as part of **FS II**.

If you think you would enjoy cruising around the countryside with **Flight Simulator II**; or if you'd like a graded set of exercises to show you how to handle your aircraft controls and instruments, you'll enjoy the first volume. The second book — more adventures — is rather weak, as though the author didn't have enough information to fill another volume and had to pad.

Try book one: it's pleasant flying. If you love the adventures, you might still want the second volume despite its weaknesses. On the other hand, maybe forty flights will be sufficient to keep you happy for quite a while. □

Microflyte Joystick System for Flight Simulator II

from Microcube

Dedicated FSII controller for Commodore 64

by Jim Butterfield

Copyright © 1986 Jim Butterfield. Permission to reprint is hereby granted, provided this notice is included in the reprinted material.

There are two 'standard' ways to fly on Sublogic's **Flight Simulator II**: use the joystick, or use keys on the keyboard. The keyboard is more precise — tap a control key a number of times for a given setting — but doesn't seem 'natural' like the control column of a real plane. A standard joystick's biggest problem is that it's a digital device: the system sees only on/off signals, and judges how far to move the controls by measuring how long a joystick postion has been held. This produces terrifying flights if you have a sticky joystick.

Now Microcube has produced a third alternative: an analog joystick that measures how far the stick has been moved. This requires both hardware and software, but can give a much more realistic feeling of aircraft control.

The joystick assembly is nicely laid out. The joystick itself is spring loaded, so that it will return to centre position by itself, or may be held off-centre by continual pressure. Additionally, the two dimensions (up/down and left/right) each have 'trimmers' that will add a permanent offset to the joystick settings. Thus, if you want to put the nose down without having to hold on to the joystick continuously, you can push the trim control forward to do the job. I wish the trim controls had a little more range to them, though.

A power light ensures that the assembly is plugged in (it connects to control port 1), and a reset button centres the controls. Well, it doesn't exactly centre them — it sets them to a takeoff attitude with the nose up. More on this in a moment. Two buttons serve as throttle controls to make it easy to move the power up or down, and two more buttons control the flaps. After you've landed, the upper flap button also serves as a brake control, making the tricky business of taxiing a little easier.

The software, **MicroFlyte ATC Joystick Driver**, is a clever preprocessor program that doesn't compromise Sublogic's program integrity. Load the MicroFlyte disk first, and it will tell you when to put in the **FS II** disk. The simulator then appears to load normally; but once you get going, the joystick box is in control.

The control mechanism is quite interesting. It's as if the joystick system were tapping on the computer's keyboard, correcting the settings. There's a feedback system involved here. If you are holding the joystick at an 'in-between' position, you'll see the flight control settings ticking back and forth between the two nearest fixed settings. That's nice, since you can get smoother, more precise controls.

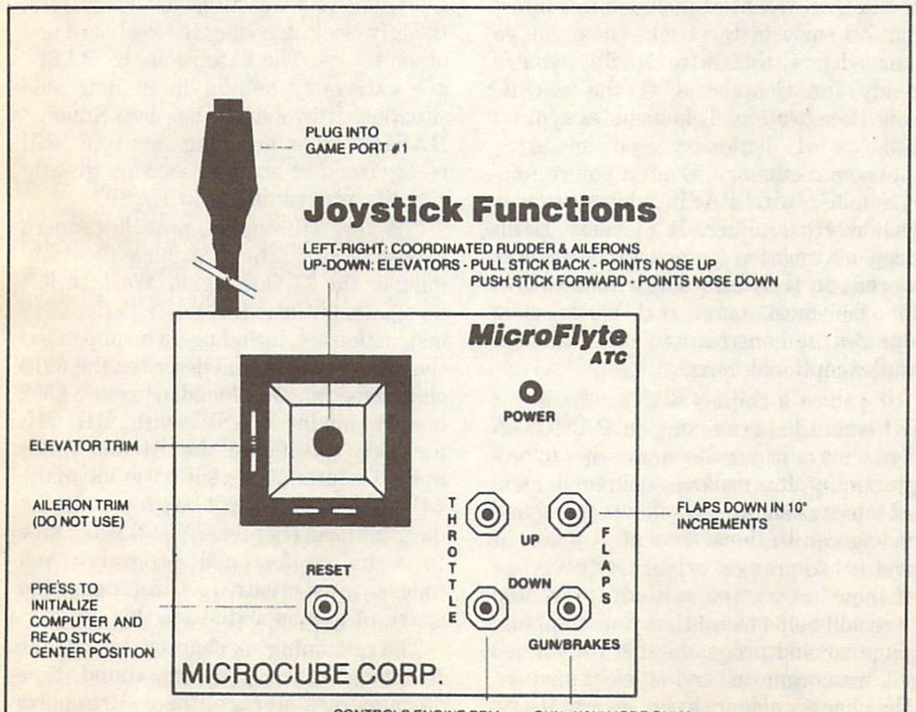
The centre area of the joystick has a deliberate 'dead spot'. You have to move the joystick a certain distance before there's any effect at all. Since the first rule of flying the **FS II** is 'gently, gently', it's useful to watch indicators 41 and 42 (Aileron Position and Elevator Position) to see precisely when your joystick movements have taken effect.

Strong movements of the joystick are to be avoided at all costs. You'll find yourself upside down, stalling, spinning or crashing (or all of the above) very suddenly if you make violent moves.

The built-in trim that the joystick gives to the aircraft is somewhat nose-high for cross-country flying, and it's hard to adjust: the trim controls won't pick up enough of the difference. But it's reasonably good for takeoffs and landings, which is most of the fun and excitement.

You can fly better using this device; it will probably increase your flying enjoyment. It's cleverly designed and well put together.

***MicroFlyte Joystick System for Flight Simulator II**, $59.95 (US), from Microcube Corporation, P.O. Box 488, Leesburg, Virginia 22075.* □


PLUG INTO GAME PORT #1
Joystick Functions
LEFT-RIGHT: COORDINATED RUDDER & AILERONS
UP-DOWN: ELEVATORS - PULL STICK BACK - POINTS NOSE UP
PUSH STICK FORWARD - POINTS NOSE DOWN
MicroFlyte ATC
POWER
ELEVATOR TRIM
AILERON TRIM (DO NOT USE)
THROTTLE
UP
DOWN
FLAPS
FLAPS DOWN IN 10° INCREMENTS
RESET
PRESS TO INITIALIZE COMPUTER AND READ STICK CENTER POSITION
GUN/BRAKES
MICROCUBE CORP.
CONTROLS ENGINE RPM UP TO INCREASE DOWN TO DECREASE
GUN - WWI MODE ONLY
BRAKES - STOP PLANE ON GROUND, AFTER LANDING AND MANEUVERING ON GROUND
FLAPS - RAISES ALL THE WAY UP.

The Walker

1) Insert your COMAL disk in drive*.
2) Type LOAD "C64 COMAL*",8
3) Type RUN (starts COMAL)
4) Type AUTO
   (COMAL provides the line numbers)
5) Enter the program lines shown below
   (COMAL indents lines for you)
6) Hit RETURN key twice when done
7) Type RUN
   Watch an animated sprite hobble across the screen. Change the (99) in line 450 for really fast walking

```
0010 setup
0020 repeat
0030   walking
0040 until key$="q" //Q to Quit
0050 //
0060 proc setup
0070   blue:=14; pink:=10
0080   white:=1; black:=0
0090   define'images
0100   repeat
0110     input "speed (1-10): ": speed
0120   until speed>=1 and speed<=10
0130   background black
0140   setgraphic 0
0150   spriteback blue,pink
0160   spritecolor 1,white
0170   spritesize 1,false,false
0180   plottext 1,1,"press q to quit"
0190 endproc setup
0200 //
0210 proc define'images closed
0220   dim shape$ of 64, c$ of 1
0230   shape$(1:64):=""
0240   shape$(64):=chr$(1)//multicolor
0250   c$:=chr$(0)
0260   for x=22 to 63 do shape$(x):=c$
0270   c$:=chr$(170)
0280   for x=1 to 21 do shape$(x):=c$
0290   define 0,shape$
0300   c$:=chr$(20)
0310   for x=22 to 42 do shape$(x):=c$
0320   define 1,shape$
0330   define 3,shape$
0340   c$:=chr$(60)
0350   for x=43 to 63 do shape$(x):=c$
0360   define 2,shape$
0370 endproc define'images
0380 //
0390 proc walking
0400   for walk:=1 to 319 div speed do
0410     x:=walk*speed
0420     y:=100+walk mod 4
0430     spritepos 1,x,y
0440     identify 1,walk mod 4
0450     pause(99)
0460   endfor walk
0470 endproc walking
0480 //
0490 proc pause(delay) closed
0500   for wait:=1 to delay do null
0510 endproc pause
```

* If you don't have COMAL yet, order a <u>Programmer's Paradise Package</u>-$19.95. It includes the complete COMAL system plus over 400 pages of information. Add $5 more to get our 20 interactive lesson Tutorial Disk. Add $2 shipping. Visa/MC or US funds check accepted. Send to:

COMAL Users Group USA
6041 Monona Drive, Room 109
Madison, WI 53716
phone 608-222-4432

Programming the Commodore 64

by Raeto Collin West

from Compute! Books

$19.95 (US), 609 pages

Review by Malcolm O'Brien

There are a lot of old-timers (PET users) who will tell you that Raeto Collin West's *Programming the PET/CBM* is the bible of green screen era Commodore computing. Now West has applied his formidable skills to the task of creating a similar volume for the C-64, and it is likely to receive the same measure of acclaim as his previous work. *Programming the Commodore 64* is an exhaustive work, weighing in at 609 pages including the index. And those are fat-free pages – just the facts with no filler. This book is chock-full of information.

West is authoritative and thorough without becoming wordy. Each topic is handled in a very concise and direct manner, often including references to other portions of the book where the reader will find associated information. These references supplement a subdivided table of contents, a good index and eighteen appendices. Finding what you're looking for is a snap.

One of the very best things about this book is the author's approach. At all times, you are made aware that you are working with a complete computer system. It's important to remember that your programs do not exist in a vacuum and that the environment must be taken into consideration if your programming is to be effective. You might write a superb accounting program, but if it has long garbage collection delays, no one will use it (not even you!). West has adopted a very sensible approach that will be appreciated both by neophytes and more experienced programmers. In fact, the latter may feel that *Programming the Commodore 64* is the very book they wish they had had when they were starting out.

Let's look at an example from the BASIC Reference Guide in Chapter 3. In covering the keywords **for ... to ... [step]**, West includes four notes on the functioning and use of **for-next** loops. The first note tells how **for-next** works and includes a line of code to demonstrate. This line "lists 18 bytes from the stack; these are the **for** token, the two-byte address of the loop variable, the step size in floating-point format, the sign of the step, the floating-point value of the upper limit of the loop, the line number of the **for**, and the address to jump to when the loop is finished." The second note is about loop execution speed; the third covers exiting from loops; and the fourth gives a method for simulating a **do-while**. This is in addition to the keyword's type, syntax, modes, token, abbreviated entry, purpose and four examples. You may learn more about **for ... to ... [step]** in these two pages than you will in a year of programming. The whole book is like this: telling you everything you need to know where and when you need to know it, instead of making you find things out the hard way.

Although the book claims to be written "just above the introductory level", I would even recommend it to beginners if they're serious about learning to program their computer. They might only be able to use a small part of it at the outset but, as they progress, they're sure to find *Programming the Commodore 64* to be a reliable companion.

Chapter 1 is a five-page introduction to the book. Chapter 2 details the 64's connectors, its keyboard and a section on editing BASIC on the 64. Introductory material often ignores editing: many beginners (myself included) have found themselves in quote or insert mode and not known what was happening. It's good to see that West has included this information early in the book. Once you've learned how to talk to BASIC, you're ready for Chapter 3 – the BASIC reference section. This includes syntax, a keyword dictionary and an error message dictionary. Even if you're fairly familiar with BASIC, this section is well worth reading. It includes 'down deep' information on such tricky subjects as **rnd**, **st**, **ti** and **ti$**. The examples are, for the most part, real world code samples demonstrating both standard and exceptional usage.

If you're a coding novice, Chapter 4 (Effective Programming in BASIC) offers some assistance in approaches to programming. It introduces such fundamental topics as algorithms, flowcharting and coding conventions. Though it does not pretend to provide exhaustive coverage of these topics, the material presented here will help ensure that your thinking, planning and programming are carried out in an organized and efficient manner. The chapter also includes several useful subroutines and short programs to perform such tasks as dice simulation, card shuffling, hex/decimal conversion and processing dates. Page 87 is a helpful list of 14 points entitled Debugging BASIC Programs.

When you get to Chapter 5, you may figure that you're in over your head. This chapter covers 64 architecture, including such heavyweight topics as: memory configurations, the 64 schematic diagram, PLA's (Programable Logic Arrays), programming the CIA's (Complex Interface Adaptors), how cartridges work, and port pinouts. Sound complicated? It is. But don't let it cause you to throw in the towel. Skim through it anyway. I thought that this chapter should have come later in the book. On the other hand, I recognize that it's important to get an early overview of memory maps and such things as buffers, pointers, vectors, flags and the stack. They may not mean much at the beginning but their significance will increase as you progress. The chapter also includes **MicroScope**, a program to display the contents of any section of memory.

Chapter 6 covers advanced BASIC: how BASIC is stored in memory, special locations and features of BASIC, and a dictionary of extensions to BASIC. Here you'll learn about line links, program chaining, storage of simple variables and arrays and the consequences of BASIC's storage methods. Although I'd heard or read about garbage collection often, I never quite understood what it was or how it worked until I read this book. Locations and Features covers buffers, the jiffy clock, decoding the keyboard and other topics. The extensions to BASIC are extremely helpful in innumerable situations. Anyone who has used **Simons' BASIC** or something similar will recognize that such extensions greatly simplify programming in BASIC.

Chapters 7 through 11 provide a superb exposition of machine language programming on the 64. Once again, West has left no silicon unturned. He details the 6510 instruction set, including (in an appendix) the quasi-opcodes, and describes the 6510 chip (something not found in generic 6502 books), mixing BASIC with ML, ML methods specific to the 64 and much more. Chapter 11 is a superb guide to the 64's ROMs. You won't be groping in the dark in these chapters. You'll learn how to write professional programs and unleash your creativity. Just supply the spark of genius and study this book!

The remaining six chapters look at the following topics: graphics, sound, tape storage, disk storage, the control ports and major peripherals. It's in this last section that you may need to supplement the information contained in this book. It will not tell you how to make your Mitey Mo hang up the phone, for instance, or which code to send to your HeadSqueuler-80 printer. But if you know what your external device requires in order to perform a specific task, West will show you how to get that (or anything else) out of your 64.

If your computer library is only going to contain one book, it should probably be this one. That's a strong recommendation but it is well earned. *Programming the Commodore 64* is first class all the way in terms of its style, approach and the thoroughness of its treatment of the 64.

Programming the Commodore 64*, by Raeto Collin West, $19.95 (US), published by Compute! Books, P.O. Box 5038, F.D.R. Station, New York, New York 10150.* □

C-128 Programmer's Reference Guide

Bantam Computer Books
744 pages, 1986
$21.95 (US), $24.95 (Cdn.)

Review by Miklos Garamszeghy

Copyright © 1986 M. Garamszeghy

The Commodore sponsored *C-128 Programmer's Reference Guide* is perhaps the most complete piece of documentation currently available for the 128. Its 744 pages are divided up into 16 main chapters and 12 appendices. Each chapter is crammed with detailed information on how to use one or another of the 128's features. BASIC, machine language, graphics, sprites, sound and the 80-column video chip are all given excellent coverage in their own chapters. CP/M rates both a main chapter and a detailed appendix. Despite the large amount of technical data included (detailed hardware specifications and schematic diagrams), the book is very easy to read and follow along with its numerous example programs.

The list of authors includes many people who worked on both the hardware and ROM software development for the C-128 project. Despite this impressive list, the book is not without its errors. While some of the errors in the explanations of the BASIC commands and control codes have been corrected from the original *System Guide*, others have not.

Admittedly, the errors are minor, such as reversing the functions of **ctrl-l** and **ctrl-k**, but they can be very frustrating for a novice who has done everything 'by the book' and cannot understand why his or her program won't work properly. Now for my favourite topic: 1571 disk drive burst mode. This very powerful aspect of the 1571 is barely mentioned in passing. Perhaps CBM thought it was adequately covered in the 1571 users' manual. Anyone who has tried to decipher the explanation of burst mode in the 1571 manual knows better.

On the positive side, the *C-128 Programmers Guide* contains very detailed (for a Commodore publication) memory maps for all three C-128 operating modes. Serious ML programmers will like this nice touch, along with the detailed explanation of C-128 memory management and the new Kernal routines. Both novice and experienced CP/M users will find the chapters detailing CP/M mode (complete with disk format explanations, CP/M BIOS and BDOS routine descriptions) very informative and interesting. Hardware hackers will enjoy the chip specifications and pin assignment diagrams. □

COMPUTE!'s 128 Programmer's Guide

COMPUTE! Books, 1986
444 pages, $14.95 (US)

Review by Miklos Garamszeghy

Copyright © 1986 M. Garamszeghy

The 444 pages of *COMPUTE!'s 128 Programmer's Guide* are divided into 7 main chapters and 6 appendices. The information is presented in a clear, easy to understand style with plenty of examples. In most respects it is quite similar to the CBM book, although with perhaps not quite so much detail.

The chapter on peripherals makes an attempt at explaining 1571 burst mode operation. While the explanation of the hardware function itself is adequate, no attempt is made to explain the software commands used to access burst mode. In addition, the example machine language program to read a file in burst mode is plain wrong and will not work properly as listed. (It will *appear* to work, but you will not be able to read the last sector of the file.) In addition, the quoted read speed of 3840 bytes per second is a bit optimistic for this type of burst mode read, which I have timed at more like 2200 bytes per second.

The COMPUTE! book does, however, have a few features lacking in the CBM book. The major one is a better section on CP/M machine language, including a complete listing of Z80 machine language mnemonics and op codes. While 65xx ML codes should be reasonably familiar to most veteran Commodore users, the Z80 codes used in CP/M mode probably are not. The description of BASIC commands and functions also includes one which is not detailed anywhere in Commodore literature: **rreg**, which is used to read the A, X, Y, and P registers of the 8502 processor after it has returned from a machine language routine. The CBM manual lists **rreg** as a reserved word, but offers no explanation as to its function. Perhaps the designers forgot about this command?

In terms of price per kg of paper, the CBM guide is by far the better deal. It appears to be written so that it can be used by programmers at all levels, from novice to advanced. While novice users may find some of the detail confusing, they will appreciate this extra detail later as they advance and become more familiar with the 128. The COMPUTE! guide, on the other hand, appears to be aimed at the novice to intermediate user with its more simplistic explanations and lower level of detail. Don't get me wrong. Both books are well written and extremely informative. Every serious C-128 user should have at least one of them. I have both. □

The black book of C-128

by Robert H. Taylor
and Dell Taylor
Holland Publishers
260 pages

Review by Miklos Garamszeghy

Copyright © 1986 M. Garamszeghy

The 'black book' is a small, spiral-bound creation, not much larger than a paperback novel, which is intended to be a quick reference tool for the C-128 and the 1541 and 1571 disk drives. Its 260 pages are divided into seven colour-coded sections, each devoted to a specific topic such as C-64 mode, C-128 mode or CP/M.

The book contains numerous summary tables and memory maps, as well as basic information on the machine and its major peripherals. In places it looks like a low budget clone of *The Transactor* magazine's indispensable *The Complete Commodore Inner Space Anthology*. I say 'low budget' because the book is not typeset, but was produced on an NLQ dot matrix printer.

While the book is certainly a handy collection, none of the information it contains is new. In fact, most of it appears to be garnered from the various Commodore instruction manuals, complete with the original errors. (I notice that the manuals are listed in the reference section, although no credit is given in the text.) For example, the list of C-128 control codes on pages 86 to 88 is reprinted from the C-128 System Guide, Appendix I, including the errors. The chapter on the 1571 drive does not even mention in passing the most powerful aspect of the device: burst mode.

While the 'black book' is a good idea in principle, it fails because it contains no more information than comes free in the manuals with the hardware when you buy it! The only semi-novel thing in the book is the last section, entitled 'Personal Data', which is approximately 30 pages of formatted blank spaces for you to record everything from equipment serial numbers to repair expenses to BBS numbers. It even includes several pages of suggested disk ID codes starting with AA and working its way up. You can tick off each one as you use it. Heaven forbid if you ever started to duplicate the ID codes on your disks! □

Out-Think

from Kamasoft

CP/M Outline Processor for the C-128

Review by Adam Herst

There are certain types of software that it is almost obligatory to own. There can't be many computer users without at least one word processor, for instance, and database managers are nearly as widespread. Such programs have become standards because of their ability to take advantage of the unique abilities of the computer and apply them to everyday problem situations. With most of the possible refinements to these programs already made, it might seem that the potential for improvement to information processing techniques has been exhausted. Not so. Prepare yourself for what is sure to become the next required piece of software.

Out-Think from Kamasoft is billed as an 'outline organizer' or 'thought processor'. Designed to facilitate the organization of thoughts in the writing process, it combines the features of a word processor and a database, resulting in a product whose abilities are greater than the sum of its parts. As a new concept in software, comparisons with existing products are difficult. Consequently this review will focus on the ideas behind the software and on the ability of **Out-Think** to perform as advertised.

Out-Think comes on a single-sided, Osborne format disk. The package includes two manuals: the *User's Guide* is a soft-cover book, while a separate, read-me-first pamphlet contains the installation instructions. Also included are two command cheat-sheets summarizing the commands of the two possible keyboard configurations.

The *User Guide* is divided into two parts: a tutorial section and a command reference. The tutorial section is excellent. Realizing that they are dealing with a new concept in software, the authors of the manual have taken the time to introduce the concepts behind outline organization. They follow this with an application tuturial that illustrates all the major capabilities of the program. The *Users Guide* includes an index for quick reference, but it is inadequate. Consequently it is difficult to find a specific piece of information when you need it.

As with many new versions of CP/M programs, the C-128 is offered as a choice on the installation menu, resulting in quick and simple installation of **Out-Think**. The installation pamphlet recommends that you make backups of the original disks. This is always a good procedure to follow, but is especially so with **Out-Think**, because the installation file erases itself during the course of installation. If you don't make a backup you will not be able to reinstall **Out-Think** in the future.

During installation you will have the choice of configuring the keyboard in two fashions: either a **WordStar** emulation or an **Emacs** emulation, whichever you are more comfortable with. This choice can be changed at any time by re-installing **Out-Think**.

Out-Think allows you organize your thoughts into topics, sub-topics, sub-sub-topics and so on, and finally text. **Out-Think** organizes ideas into three groupings. At the highest level is the topic. Structurally, a topic is a file on the disk. Underneath topics are various levels of sub-topics. At the bottom of the heap is the text leaf. The latter two exist within a topic disk file. While not difficult to grasp, this organization is cloaked in a unique terminology that some users might find confusing at first. Nonetheless, there is method behind the madness that imposes a logical order on the naming convention.

Out-Think has three modes of commands corresponding to the three levels of organization. The primary interface is the topic manager, which allows manipulation of the overall **Out-Think** environment. Using the topic manager, you can manipulate and create topics in their entirety as well as copying and resizing existing topics.

At the next level down is the Topic Editor. This editor, as its name implies, allows the creation and manipulation of sub-topics within a topic file. In the topic editor you can promote or demote a sub-topic in importance, extract, copy and modify existing sub-topics into new sub-topics or even into new, unique topics. Also available are various types of elementary search and find commands to quickly locate strings embedded at any level of precedence. As well as the standard text string match, the search function also operates on a phonetic level. This means that a search can be successful even if the search string is misspelled or inaccurate.

At the lowest level is the leaf editor, which allows the entry and manipulation of textual information within a topic file. In function, the leaf editor is similar to a word processor, and offers similar functions, including a full screen editor, block operations and formatted printer output.

Each mode has a unique set of commands associated with it. A large number of commands exist for each mode. Across modes, the commands attempt to remain consistent in the operations they invoke. Some of the commands are single keystrokes, while others involve a series of **ctrl** and **esc** codes. As an option is chosen, subsequent options are displayed in a status line at the bottom of the screen. At least one help screen is available in each mode by pressing various combinations of the **esc** and **help** keys. Unfortunately, these help screens are merely online versions of the distributed cheat-sheets, and do not offer information relevant to a specific problem. (Interestingly, the help files are themselves an Out-Think topic file.)

Once you have entered and organized your information in the desired format you can print either all or part of the resulting document on a printer, or write it to a disk file to further refine it in a word processor. A variety of formatting commands are available to control output, including line spacing, margin control, header and footer information, table of contents, and **WordStar** file output. Alternatively, text files can be imported into **Out-Think** and transformed into an outline format (branches, stems and leaves).

You may be wondering just what you would do with a product like **Out-Think**. In the same way, you may once have wondered what you could do with a word processor. How indispensable has that word processor become for you since then? **Out-Think** or another outline processor is likely to assume similar importance in your life. Even though it contains aspects of many popular types of programs, **Out-Think** proves to be greater than the sum of its parts. I have used it for all sorts of projects. As an organizer for essays and articles, **Out-Think** replaces the old pen and paper outlines. From the outline, the text editor can be used to produce a more than adequate rough draft. The ability to organize and do retrieval searches lets **Out-Think** be used as a simple database. In fact, nearly anything that involves organization, retrieval and writing can be done effectively using **Out-Think**.

As with any piece of software, **Out-Think** is not without its problems. Unfortunately, most of the problems can be traced back to the characteristics of CP/M on the C-128, and are not inherent in the software. (This is unfortunate, because problems in the software can always be remedied, while the remedy for hardware problems is usually to buy a new computer).

Most CP/M programs are disk intensive. This means that the program, the data or both are stored on disk and brought into the computer only when needed. Unfortunately, **Out-Think** is *very* disk intensive. While the program itself is memory resident, the data on which it operates (the topics, subtopics and text leafs) are all stored on disk and called into memory only as required. While the C-128/1571 combination offers a great improvement in speed over other Commodore computers, it still falls short of the speed that most other CP/M machines are capable of. Consequently, **Out-Think** operations involving disk access (that is, most **Out-Think** operations) can be excruciatingly slow. This is one program that will benefit greatly from the availability of a RAM disk.

Nonetheless, even without the RAM disk, my recommendation for this product is an unqualified 'buy it!'. This is one of the most useful and most used programs I own. It has become so indespensable to me that it is practically the only CP/M program that I am using. It works as advertised, and I have yet to find any bugs. If that isn't enough to convince you to buy it, consider these last two points: **Out-Think** costs only $49.00 US and is not copy-protected. Can't this company do anything wrong? □

COMPUTE!'S VIC 20 Collection

from COMPUTE! Books
335 pages, $12.95 (US)

Review by Roger Burge

According to its cover, the articles in this volume have not been published before. However, many of them *have* appeared, in one form or another, in magazines from the same publisher.

Nevertheless, this is a good colection of material for VIC users at the beginner or intermediate level. Some intermediate users may see some of the programs as starting points for their own creations, while both groups will find the utility section quite useful.

Tutorial subjects include custom characters, relative files and memory conservation. These subjects will be appreciated by the novice and experimenter alike.

Utilities include screen scrolling, printer dumping, a micro-assembler, joystick decoding and a number of other helpful ideas. Many of them include machine language subroutines that speed things up considerably compared to similar BASIC routines.

While the home applications chapter is quite weak, the game section is not. Some of these games have been done better (and are available from TPUG under various titles); the others range from fair to excellent. Almost forty games are featured.

You certainly won't like all the games, nor will you find use for all the utilities, but there is enough material covered to please a wide group of VIC users. The well-printed listings will permit you to customize the BASIC programs, and some may find themselves tempted to fool around with the machine language subroutines too.

As with most COMPUTE! books, the writing style is concise and easy to follow or learn from. Unfortunately, as with their magazines, some program listings include typographical errors that can drive you crazy.

This is definitely a good purchase for VIC owners who enjoy learning and experimenting. □

VIC 20: Easy Guide To Home Applications

by Harold O. Fisher

VIC 20 Games'n'More

by Earl R. Savage

both from Howard W. Sams
$8.95 (US) each

Review by Roger Burge

These two books are published for the novice computer user. Unfortunately, even new VIC users may be disappointed.

Games'n'More promises, on its back cover, to "expand the use of this machine beyond ... what is available in prerecorded programs ... inexpensively".

At $12.95 (Cdn.), however, this is not a software bargain. While the author does explain his programming techniques as promised, and while this can be helpful to novices, the average person will be able to get as much of a head start by reading such magazines as *COMPUTE!'s Gazette*.

The games presented here are the same ones that have been floating through the public domain for years: **Battleship**, **Tic-Tac-Toe**, **Depth Charge**, and so on. Virtually any of them can be had from TPUG's VIC 20 library in a much more polished and entertaining form. Little use is made of the VIC's fine sound and graphics features, which are often what attracts people to a home computer.

Easy Guide To Home Applications suffers from the same problems. More useful applications abound in the public domain. Moreover, some of the applications are not very useful — they can be performed more easily with paper and pen. Programs like **Your Child's Height** and **Jogger's Log** do not require a personal computer's power. The applications that *are* more appropriate to a computer are sadly lacking in power compared to similar exercises in TPUG's own library. □

The Halley Project

from Mindscape, Inc.

Educational game for Commodore Amiga

Review by Tim Grantham

The Halley Project: A Mission In Our Solar System is a victory of style over substance. This is by no means a problem if the intended audience is of the right mental age. And **Halley Project** does have good style.

It all starts with some very slick packaging – a *Mission Reference Guide* contained in a dossier marked 'Top Secret', a flight instructions reference card, a 'Simple Star Map', and a cassette tape containing a recording of the mission briefing from P.L.A.N.E.T. HQ.

Booting the game disk, which is copy-protected, brings up a very impressive opening sequence, complete with animated credits a la *Star Wars* and a digitized recording of Tom Snyder, designer of the program, singing the theme song. Unfortunately, this auspicious beginning is not followed through in the program itself. While the graphic content is far superior to the C-64 incarnation of **Halley Project**, the designers have not really taken advantage of the Amiga's superb graphics or sound capabilities.

All ten missions begin and end at Halley's comet. Your objective is to complete each mission in the shortest possible time. Each mission consists of several 'legs': for example, one might be instructed to fly first to Mercury, then to Titania (one of the moons of Uranus), then to Earth's Moon and, finally, back to Halley's comet. You don't find out where the next destination is until each leg is completed. Each time you land, you are rewarded with a view of the surface. These are rather well done, and are an indication of what could *really* have been accomplished with the graphics. They make one wish the designers had included planetary features viewable from space, rather than the blank coloured orbs they are now. Saturn doesn't even have *rings*, for heaven's sake (pun intended)!

The entire game is played with the mouse – a good feature, I think, for the small hands of children sometimes have problems with joysticks. To navigate, you bring up the radar screen, which presents you with a 'bird's-eye' view of the Solar System. From here, you determine the direction and distance of the destination. It might be 300 million kilometres away, half way between the constellations Taurus and Aries. You then rotate the ship so that it is aimed between these two constellations in the viewport and accelerate. Pressing the right mouse button brakes the ship when you are in the vicinity of your destination. A 'hyperwarp' feature is provided so that you don't grow old and die while travelling to Pluto!

The mission destinations are not usually spelled out for you. They are given in the form of clues – "Your destination is any moon larger than Titania..."; or "Your destination is any planet colder than Jupiter." These are sometimes frustratingly vague: "Any planet that is smaller and *always* colder than Earth" (my emphasis) should strictly speaking rule out Mars as a possibility – temperatures at the equator of Mars sometimes reach 85 degrees Fahrenheit. However, **Halley Project** accepts both Pluto and Mars as answers. Presumably, it is average planetary temperature that's intended in the clue, but this is not made clear, either in the program or the documentation. Also, alert students may be tripped up by the fact that **Halley Project** has Pluto as the ninth planet; it is, in fact, currently the eighth because its highly eccentric orbit actually crosses inside that of Neptune's.

None of the clues is particularly challenging: even young children need only refer to a very basic astronomy book to solve them (as they are encouraged to do by the documentation). And some compromises in accuracy have been made. The designers note that current microcomputer technology prevents displaying the stars at their correct magnitude (brightness); and that to enhance playability, all planets and moons have been moved to the plane of the ecliptic. I don't agree with either of these decisions, as regards the Amiga version. The real reason, in my view, was simply that to include these capabilities would have meant spending a little more time in development, time that Mindscape perhaps felt could be better spent earning money in release.

Still, flying around Jupiter while dodging the moons does give a real sense of the spatial relationships involved, because all moons and planets are moving in their actual orbits as the game runs.

The music and sound effects during play quickly became irritating: the two bars of music endlessly repeated whenever I was within 100,000 kilometres of a planet or moon soon lost its charm; and the hyperwarp warning sound went through my head like drill.

Landing is the only tricky part of the game. Once you are within 100,000 kilometres of your destination, you can receive a signal from the landing beacon. Once that has been detected, the automatic landing sequence can be commenced. However, detecting the signal of the beacon sometimes means circling the heavenly body in question until it is found. As no attempt has been made to simulate the effect gravity has on the ship, you cannot simply place your ship in a convenient orbit and wait. You must actually 'fly' it around the planet or moon. As ballistic trajectories are a rather more advanced topic than the average student can probably handle, this is an acceptable compromise.

Provision is made for a number of different players to store the results of their flights. As times improve, comparisons can be made between players. The times and positions are saved to the program disk, something I find a little disturbing, being by nature cautious. But others may not share my concern.

I look forward to a second generation version of **Halley Project** – one that would truly exploit the Amiga's capabilities. I can't think of a better idea for an educational program than an opportunity to explore the neighbouring cosmos.

For a limited time, Mindscape is making available a secret eleventh mission to those who complete the ten missions contained in **Halley Project**. As soon as I find out where Iapetus is, I'll be on my way.

***The Halley Project: A Mission In Our Solar System**, $49.94 (Cdn.) from Mindscape, Inc., 3444 Dundee Rd., Northbrook, Illinois 60062.* □

Carriers At War
from Strategic Studies Group
Naval combat simulation for Commodore 64

Review by Dave Dempster

World War II, at sea, in the Pacific — the mix of sea room, limited (and suspect) tactical intelligence, and fast concentration against parts of the enemy's fleet — has all the elements to produce an exciting and interesting game. Two Australian gentlemen, Roger Keating and Ian Trout, have addressed this challenge in **Carriers At War**, a Strategic Studies Group simulation, and it's a beaut!

The game offers several scenarios from Pearl Harbour to the Philippine Campaign. It is playable both against the computer (your best option) and against another player (though you are not allowed to peek when the other player is working on a move). It also offers the intriguing option of the computer playing against itself, and even lets you choose to control the forces at sea while the computer handles your land based air forces or other fleets.

The extensive menu system is layered four deep. It is logically organized going from general to specific and, though it looks formidable, works astonishingly well. You can step to any menu that your command status permits you to access, and input commands for Task Group or strike/ready commands to aircraft squadrons. You can also receive intelligence reports, weather information, Task Force status and even individual ship/squadron status.

Once you have given orders and garnered information, you sit back and watch the action unfold on a large-scale strategic map showing, naturally, only your units or spotted enemy units. An hour passes in about 15 seconds in the game (the game clock shows 10 sec intervals) and the game can be interrupted for input at any time. The game will automatically stop if a significant action occurs (such as spotting an enemy unit or being spotted by one). The game becomes particularly interesting when you hear the sound of aircraft launching yet nothing shows on the screen. Raids don't show until the radar of your units would have spotted them.

The sounds in the game consist of strange 'plunks' and 'whees' that do become useful once you understand what they signify. The graphics are serviceable and the menu displays are well done. The dots showing the centres of each hex on the large scale map are useful for planning (the tactical display shows hexes 20 miles across). The manuals are explicit, complete and concise.

One feature of **Carriers At War** that places it head and shoulders above any other game I have seen is the built-in capability to modify scenarios, or to create your own. In fact, the data is presented for a complete extra game called 'Raid on Ceylon', along with detailed directions on how to build the game. This procedure is not for the faint-hearted: it involves considerable time on the keyboard. Many details of ship and aircraft performance, morale, equipments, technical specifications, and even weather must be specified. The capability of modifying available scenarios permits you to play Pearl Harbour with an aroused USN or, more interestingly, design four variants on a game such as Coral Sea, then have someone else pick the scenario so you're unaware of the size of the opposition. The game designer's kit is, by itself, worth the price of the game.

Carriers At War is a joy to play because it calls on you to make the big decisions but automatically handles routine and tedious tasks such as turning into the wind to launch aircraft, launching CAP during daylight hours, and verifying that the aircraft you launch are within range of their targets. You designate search quadrants for fleets or bases reconnaissance capability, and the quadrants are searched in accordance with the skill and number of aircraft available. The result is that **Carriers At War** plays fairly fast. I also found that the design encouraged me to think in broad strategic terms by relieving me of the need to worry about distracting details.

What didn't I like about **Carriers At War**? The time required to access the disk between game modes can be mildly annoying, though the fault lies more with the 1541 disk drive than the game itself. Also, you must reboot to switch between Create mode and Game mode. My copy occasionally crashed in the middle of the Create routine, which is very disconcerting. The authors say to save a developing game frequently — good advice.

I rate **Carriers At War** as among the best of the new type of simultaneous, limited-intelligence games. It's exceptional. My answer to the decisive question — "Now that you know it, would you still buy it?" — is a loud 'Yes!' □

Missing Letter Puzzle

1) Insert your COMAL disk in drive*.
2) Type LOAD "C64 COMAL*",8
3) Type RUN (starts COMAL)
4) Type AUTO (turn on auto line#'s)
5) Enter the program lines shown below (COMAL indents lines for you)
6) Hit RETURN key twice when done
7) Type RUN
   Pr-gr-mm-r's P-r-d-s- P-ck-g-
   Programmer's Par?

```
0010 dim text$ of 39, disk$ of 2
0020 open file 2,"missing.dat",read
0030 disk$:=status$; count:=0
0040 if disk$="00" then
0050   count'text
0060 else
0070   close // no data file found
0080   create'text
0090 endif
0100 play'game
0110 //
0120 proc count'text
0130   while not eof(2) do
0140     read file 2: text$
0150     count:+1
0160   endwhile
0170   close
0180 endproc count'text
0190 //
0200 proc create'text
0210   open file 2,"missing.dat",write
0220   print "input text (or blank):"
0230   repeat
0240     input text$
0250     if text$>"" then
0260       write file 2: text$
0270       count:+1
0280     endif
0290   until text$=""
0300   close
0310 endproc create'text
0320 //
0330 proc play'game
0340   open file 2,"missing.dat",read
0350   for x:=1 to rnd(1,count) do
0360     read file 2: text$
0370   endfor x
0380   close
0390   for letter:=1 to len(text$) do
0400     if text$(letter) in "aeiou" then
0410       print "-",
0420     else
0430       print text$(letter),
0440     endif
0450   endfor letter
0460   print
0470   for letter:=1 to len(text$) do
0480     while key$<>text$(letter) do
0490       print "?"+chr$(157), //left
0500     endwhile
0510     print text$(letter),
0520   endfor letter
0530 endproc play'game
```

* If you don't have COMAL yet, order a Programmer's Paradise Package-$19.95. It includes the complete COMAL system plus over 400 pages of information. Add $5 more to get our 20 interactive lesson Tutorial Disk. Add $2 shipping. Visa/MC or US funds check accepted. Send to:

COMAL Users Group USA
6041 Monona Drive, Room 109
Madison, WI 53716
phone 608-222-4432

Products Received

Presented by Astrid Kumas

The following products have been received by TPUG Magazine *in recent weeks. Please note that these descriptions are based on the manufacturers' own announcements, and are not the result of evaluation by* TPUG Magazine.

Kermit's Story Maker

Kermit's Electronic Story Maker, from Simon & Shuster, Inc., 1230 Avenue of the Americas, New York, New York 10020. Price $39.95 (US).

Kermit's Story Maker for the C-64 is introduced to prospective buyers as "a spin-a-word writing kit for beginning readers ages four and older". This description could be considered misleading by those parents whose main concern is to develop their chidren's writing skills. **Kermit's Story Maker** does not involve children in actual writing: they create their one-sentence stories through the use of a joystick. What the program does do is provide an opportunity to develop a sight-reading vocabulary. As fast as children can place a word on the screen with their joystick, they will see it illustrated. The designers of the program use all of the capabilities of the computer — colour, sound and animation — to make the learning experience both enjoyable and fruitful.

In **Kermit's Electronic Story Maker** the computer screen represents a blank page with a set of white lines at the top for words. A young player puts together a sentence by moving the cursor up to these lines and pressing the joystick button to spin through a selection of words. As sentences are built, the main character, the action and the location are determined. At the same time, a picture appears in the lower half of the screen: it changes as the author selects new words to fill the blank lines. Sound effects change too. Each location has its own special music — very well done indeed!

While going through the pages of their stories, children encounter different sentence structures. They are limited in number, and after a while may become too repetitive. However, before that happens, children will spend some time experimenting with different combinations of words and, undoubtedly, acquire some useful language skills.

The program allows children to flip backward and forward through the pages of their stories, save them onto a separate disk, read them later page by page and erase them if they do not seem interesting any more.

Keyboard Cadet

Keyboard Cadet from Mindscape Inc., 3444 Dundee Road, Northbrook, Illinois 60062. Price: $39.95 (US).

Keyboard Cadet is a self-paced typing course for Commodore 64 users. The program teaches touch-typing on both QWERTY (standard) and Dvorak keyboards. Designed as an arcade-style game, **Keyboard Cadet** is easy enough for young children to master. However, it will also appeal to teenagers and experienced typists, who should find the three levels of difficulty quite challenging.

While on the space mission, the user undergoes the Basic Cadet Training Program. The goal is to learn to operate the control panel (computer keyboard) in order to navigate and protect one's spacecraft succesfully. The training program consists of fifteen lessons. The first twelve lessons follow a similar pattern: each lesson begins with practice at typing letters (and/or numbers and symbols), then moves on to two-letter combinations and finally words. In Lesson 13 users type words; in Lesson 14 the focus is on sentences; and in Lesson 15, paragraphs. At the end of Lesson 15, the program displays a words-per-minute rate.

An important feature of the **Keyboard Cadet** is that it teaches proper hand positioning. During the first thirteen lessons, a picture of the keyboard and a pair of hands showing correct finger position is displayed on the screen. As each letter, number or symbol is displayed, the hands move to show the proper finger reaches, and the correct key is highlighted on the picture of the keyboard.

The **Keyboard Cadet** package contains a program disk, a back-up disk, a Reference Card, a user's guide and the teacher's manual.

Aprospand-64

Aprospand-64 from Aprotek, 1071-A Avenida Acaso, Camarillo, California 93010, (805) 482-3604. Price: $29.95 (US). Insured shipping to Canada is $6.00.

Aprospand-64 is a four-slot expander for the Commodore 64, Plus/4 and Commodore 128. It allows the user to install up to four cartridges and use them independently or in any combination allowed by the function of each cartridge. **Aprospand-64** has a push-button reset switch, allowing a restart without having to power the computer off then back on. Also, the computer's power line to the cartridges is fused to protect the computer from faulty cartridges. The product carries a full-year parts and labour warranty.

Saver Switch

Saver Switch from Value-Soft Inc., 9513 S.W. Barbur Bld., Dept. 56, Portland, Oregon 97219, (503) 246-0924. Price: $29.95 (US). Add $2.00 (US) for shipping to Canada.

Serial Switch is a serial junction box with two inputs and one output, allowing manually selectable input from two sources (although only one source may be on line at a time). About the size of a pack of cigarettes, the box can be placed anywhere in the daisy chain. □

COMP-U-TEMP™

ATARI 800/XL, 130XE, & COMMODORE 64

TEMPERATURE MONITORING AND DATA ACQUISITION SYSTEM

FEATURES include display of 8 or 16 temperature channels, range of -15°F to +180°F at approx. 1 degree resolution, electronic interface plugs directly into the joystick port, weather-protected sensors, menu-driven software capable of (1) labeling sensor locations (2) selecting high or low alarm set points (3) hardcopy printouts (4) selection of data sample time intervals for all channels (15 seconds to 4 hours) (5) recording temperature data to disk (optional).

VERSION 1.0 8 Channel $99.95
Includes 2 sensors, software, electronic interface, and hard copy

VERSION 2.0 8 Channel $119.95
Includes 2 sensors, software, electronic interface, hard copy, and data storage to and from disk

VERSION 3.0 16 Channel $179.95
Includes 4 sensors, software, electronic interface, hard copy, and data storage to and from disk

Additional sensors (Each) $5.75

ATI Applied Technologies, Inc.
Computer Products Div.
Lyndon Way, Kittery, ME 03904

M/C - VISA accepted (207) 439-5074

U.S. Dept. of Energy Award for Energy Innovation.
Dealer & Distributor Inquiries Invited

Bulletin Board

New Products on Quantum Link

Robert Baker of Baker Enterprises has announced the creation of a new feature that he is managing on the Quantum Link telecommunications network. This new area was specifically set up for companies to announce information on their products for any Commodore computer system. Manufacturers can now communicate directly with Commodore users and provide them with up-to-date information about their latest products.

If you have access to Quantum Link, you can post your own product description in the special message board provided. Otherwise, your printed new-product announcements and follow-up information can be mailed to Robert W. Baker, 15 Windsor Drive, Atco, New Jersey 08004. All material will typically be posted on the system within 48 hours after it is received. If you would like to send your information in early but delay the announcement until a specific date, please be sure your request is clearly documented.

The New Product Information message board is located in the *Meet the Press* section of the *Commodore Information Network*. Just look for *Robert Baker* in the *Meet the Press* menu. Quantum Link is planning to create a new expanded section specifically for *New Product Information* later this summer. This information will then be categorized for easy access as more companies participate and the amount of information begins to grow.

Greater Omaha Commodore Users Group

The Greater Omaha Commodore Users Group, the largest Commodore users group in its region, has recently moved to another location. The new address is: The Greater Omaha Commodore Users Group, P.O. Box 241155, Omaha, Nebraska 68124.

1986 Midwest Commodore Conference/Expo

The 1986 Midwest Commodore Conference is to be held on August 9th and 10th at the Holiday Inn Convention Center, 72nd and Grover in Omaha. There will be over thirty workshops and both local and national vendor support. Speakers include:

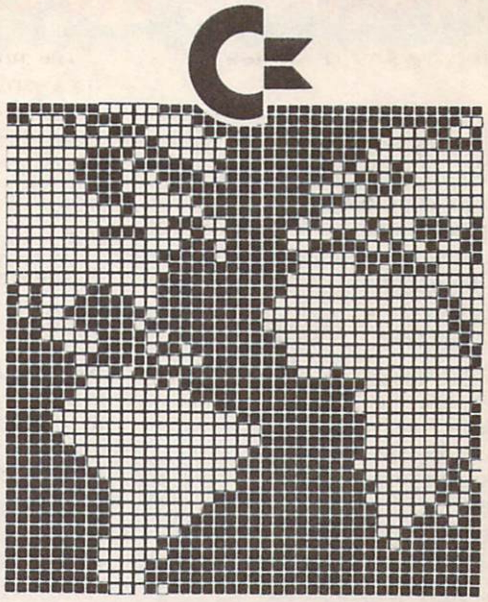
- **Jim Butterfield**, Mr. Commodore from Toronto, Ontario,
- **Dr. Richard Immers**, author of *Inside Commodore DOS, Detroit, Michigan.*
- **Valerie Kremmer**, computer language expert, Los Angeles, California,
- **Dr. James Alley**, art professor (Amiga), Savannah College of Arts and Design, Savannah, Georgia,
- **Pete Baczor**, Commodore Users Group Coordinator.

Unclassifieds

This space is for the ads of TPUG members. Wanted or for sale items only. Cost is 25 cents per word. No dealer ads accepted.

$100.00 or Best Offer. 4-year 4032 with Datasette. Warren Davies. 484-0707.

THE WORLD OF COMMODORE



The 1985 Canadian World of Commodore show was the largest and best attended show in Commodore International's history. Larger than any other Commodore show in the World and this year's show will be even larger.

World of Commodore is designed specifically to appeal to the interests and needs of present and potential Commodore owners.

Everything about your present or future Commodore computer—from hardware to software, Business to Personal to Educational. Price of admission includes free seminars, clinics, contests and free parking.

INTERNATIONAL CENTRE

DECEMBER 4-7 '86

TORONTO

Plan To Exhibit

A HUNTER GROUP PRESENTATION
For more information call:
(416) 595-5906

advertisers' index

Applied Technologies	46
Brantford Educational Services	31
COMAL Users Group, USA	39,45
Computer Rentals	21
John Dunlop & Associates	11
Electronics 2001	3
The Guide	IBC
Hunter-Nichols	47
Kobetek Systems Ltd.	11
Lattice, Inc.	BC
Magazine Back Issues	IFC
Midnite Software Gazette	11
North Ohio Firmware	31
PRGS	30
Rainbow Electronics	11
Rich-Hill Telecom International	15
TPUG (Disk Subscriptions)	25
TPUG (Conference)	21
The Transactor	31
Wilanta Arts	9

TPUG Magazine Distributors

***Dealers**: If you would like to carry **TPUG Magazine** in your store, you may order from any one of the following distributors:*

CANADA

Master Media, Oakville, Ont.	416-842-1555
Ingram Software, Concord, Ont.	416-665-0222
Compulit Distributors, Port Coquitlam, BC	604-464-1221

USA

Prairie News, Chicago, IL	312-384-5350
Levity Distributors, North Hollywood, CA	818-506-7958
Whole Life Distributors, Englewood, CO	303-761-2435
M-6 Distribution, Houston, TX	713-778-3002
The Homing Pigeon, Elgin, TX	512-276-7962
Fred Bay News Co., Portland, OR	503-228-0251
Alonso Book & Periodical, Alexandria, VA	703-765-1211
Cornucopia Distribution, Seattle, WA	206-323-6247
Guild News, Atlanta, GA	404-252-4166
Micro-PACE, Champaign, IL	800-362-9653
Nelson News 4651 F Street, Omaha, NE 68127	
Michianna News, Ft. Wayne, IN	219/484-0571
Total Circulation, South Hackensack, NJ	201/342-6334

TPUG Contacts

TPUG OFFICE 416/445-4524
TPUG BBS 416/273-6300
TPUG MEETINGS INFO 416/445-9040

Board of Directors

President	Chris Bennett	c/o 416/445-4524
Vice-President	Gerry Gold	416/225-8760
Vice-President	Carl Epstein	416/492-0222
Recording Sec.		
	David Bradley	c/o 416/445-4524
	Richard Bradley	c/o 416/445-4524
	Gary Croft	416/727-8795
	Mike Donegan (evgs.)	416/639-0329
	John Easton	416/251-1511
	Keith Falkner	416/481-0678
	Anne Gudz	c/o416/445-4524
General Manager	Bruce Hampson	416/445-4524

TPUG Magazine

Publisher	Bruce Hampson	416/445-4524
Editor	Nick Sullivan	416/445-4524
Assistant Editors	Tim Grantham	416/445-4524
	Adam Herst	416/445-4524
Production Manager	Astrid Kumas	416/445-4524
Ad Sales	Bruce Hampson	416/445-4524

Meeting Co-ordinators

Brampton Chapter	Jackie Bingley	c/o 416/445-4524
C-64 Chapter	Keith Faulkner	416/481-0678
COMAL Chapter	Donald Dalley	416/742-3790
	Victor Gough	416/677-8840
Communications	Darrell Grainger	c/o 416/445-4524
Eastside Chapter	Nina Nanan	c/o 416/445-4524
Hardware Chapter	Frank Hutchings	c/o 416/445-4524
SuperPET Chapter	Gerry Gold	416/225-8760
VIC 20 Chapter	Anne Gudz	c/o 416/445-4524
Westside Chapter	John Easton	416/251-1511
	Al Farquharson	519/442-7000
Business Chapter		
New Users Chapter		
C-128 Chapter	Adam Herst	c/o 416/445-4524
Amiga Chapter	Mike Donegan (evgs.)	416/639-0329

Librarians

COMAL	Victor Gough	416/677-8840
PET	Mike Donegan (evgs.)	416/639-0329
SuperPET	Bill Dutfield	416/224-0642
VIC 20	Richard Best	c/o 416/445-4524
Commodore 64	Derick Campbell	416/492-9518
B-128	Paul Aitchison	c/o 416/445-4524
Amiga	Mike Donegan (evgs.)	416/639-0329
C-128	Adam Herst (CP/M)	c/o 416/445-4524
	James Kokkinen (C-128)	c/o 416/445-4524
MS/DOS	Colin Justason	c/o 416/445-4524

TPUG Bulletin Board

Sysop (voice, weekdays)	Sylvia Gallus	c/o 416/896-1446
Assistant Sysop	Steve Punter	c/o 416/896-1446

The Guide

A Monthly Publication For Commodore™ Owners

Commodore support with a twist . . . Personable and even humorous . . . Timely news . . . Helpful tutorials . . . On-going support for several languages: BASIC (including BASIC 7.0 as featured in the 128 PC), Machine Language, COMAL, and Pascal . . . Program Listings . . . *Honest* software reviews.

The Guide features some of the best computer humorists to be found.

- Introduce your "widow" to the Computer Widow's Compendium.
  - Tutorials and feature articles by the famous Mindy Skelton.
    - Featuring Shelly Roberts' "I'm Sorry . . . But I Don't Speak Hexidecimal." Discover why Shelly just may be the Andy Rooney of the computer world!

We feel we have assembled one of the most talented staff of writers in the Commodore world. Receive each month the most friendly and helpful Commodore publication available. Written by Commodore users who are writing to you, not down at you.

Limited offer — FREE With Each Subscription !

Subscribe *NOW* to capitalize on free software offer!

Receive up to $99.85 in *high quality* software!

With each year's subscription (or renewal) ordered, receive your choice of the software packages listed below, including the award winning educational games from Disney, or Omiterm Terminal written by our own Bob Richardson.

Subscription	Rate	Canadian*	Receive FREE
One Year	$18.00	$24.00 (U.S.)	Any software title listed below
Two Years	$35.00	$47.00 (U.S.)	Any two of the titles below
Three Years	$48.00	$64.00 (U.S.)	All three of the titles below

* Note: Canadian (and other foreign) subscribers must add $2 (U.S.) per program title to cover shipping and handling. No handling charge for U.S. subscribers. Canadian rates listed in U.S. dollars — send U.S. funds only, please.

Donald Duck's Playground

CES Software Showcase Award Winner! — Disney animation at its best! Children play four games to "earn" money to buy playground equipment. Builds money handling skills. Superb graphics. A bestseller! $39.95 retail value.

Winnie The Pooh In The Hundred Acre Wood

— Players explore the Hundred Acre Wood to find lost articles like Ow'ls books, Pooh's honey pot and Eeyore's tail, and return them to their rightful owners. Cheery music from the Disney movie caps off this computer rendition of the beloved classic. $39.95 retail value.

Omiterm Terminal Program

— Written by *The Guide's* own Bob Richardson. Fully supports the new 1660 Modem 300! (The *ONLY* commercial terminal package that currently fully supports the 1660!!!

- **Modem controls accessible from the keyboard**
- **Punter protocol — upload & download — 300/1200 baud**
- **Ten programmable function keys**
- **15 number phone directory**
- **20k receive buffer**
- **Tone or pulse dialing**
- **Auto dial/re-dial**
- **Half/full duplex**

A $19.95 retail value.

The Guide
3808 S.E. Licyntra Ct.
Portland, OR 97222

Don't delay — mail today! Supplies are limited.

Name____________________

Address____________________

City, State & Zip____________________

Enter My Subscription for:

☐ **1 Year** ☐ **2 Years** ☐ **3 Years**

Please Check One: ☐ **New** ☐ **Renewal**

You are entitled to one FREE software package, as described above, for each year's subscription purchased. Please list your second choice, as supplies are limited on the Disney titles! Please allow six to eight weeks for delivery.

My first choice is:____________________

____

My second choice is:____________________

____

Software designed for AMIGA.

Lattice C Compiler ***$149.95***

With more than 30,000 users worldwide, Lattice C Compilers set the industry standard for MS-DOS software development. Lattice C gives you all you need for development of programs on the AMIGA. Lattice C is a full implementation of Kernighan and Ritchie with the ANSI C extensions and many additional features.

AMIGA C Cross Compiler ***$250.00***

Allows AMIGA development on your MS-DOS system. Price includes the above product.

Lattice Screen Editor (LSE™) ***$100.00***

Designed as a programmer's editor, *Lattice Screen Editor (LSE)* is fast, flexible and easy to learn. *LSE's* multi-window environment provides all the editor functions you need including block moves, pattern searches and "cut and paste." In addition, *LSE* offers special features for programmers such as an error tracking mode and three Assembly Language input modes. You can also create macros or customize keystrokes, menus, and prompts to your style and preferences.

Lattice dBC III Library™ ***$150.00***

The *dBC III library* lets you create, access and update files that are compatible with Ashton-Tate's dBASE system. *dBC III's* C functions let you extend existing dBASE applications or allow your users to process their data using *dBC III* or dBASE III.

Lattice Make Utility (LMK™) ***$125.00***

An automated product generation utility compatible with UNIX Make, *Lattice Make Utility (LMK)* lets you rebuild complex programs with a single command. Once you specify the relationships of the various pieces of your system in a dependency file, *LMK* automatically rebuilds your system the same way every time, and only compiles program files that have changed. But *LMK* is not limited to updating programs. You can use *LMK* to update documentation or perform **any** executable command!

Lattice Text Utilities™ ***$75.00***

Lattice Text Utilities (LTU) consists of eight software tools to help you manage your text files. GREP searches files for the specified pattern. DIFF compares two files and lists their differences. EXTRACT creates a list of file names to be extracted from the current directory. BUILD creates batch files from a previously generated file name list. WC displays the number of characters and optionally the checksum of a specified file. ED is a line editor which can utilize output from other *LTU* software in an automated batch mode. SPLAT searches files for a specified character string and replaces every occurrence with a specified string. And FILES lists, copies, erases or removes files or entire directory structures which meet the specified conditions.

Lattice Unicalc® Spreadsheet ***$79.95***

Unicalc is a simple-to-operate program that turns your AMIGA computer into an electronic spreadsheet. Using *Unicalc* you can easily create sales reports, expense accounts, balance sheets, or any other reports you had to do manually.

Unicalc offers the versatility you've come to expect from business software, plus the speed and processing power of the AMIGA.

• 8192 row by 256 column processing area • Comprehensive context-sensitive help screens • Cells can contain numeric, algebraic formulas and titles • Foreign language customization for all prompts and messages • Complete library of algebraic and conditional functions • Dual window capabilities • Floating point and scientific notation available • Complete load, save and print capabilities • Unique customization capability for your every application • Full compatibility with other leading spreadsheets.

Lattice MacLibrary™ ***$100.00***

The *Lattice MacLibrary™* is a collection of more than sixty C functions which allow you to quickly and efficiently take advantage of the powerful capabilities of the AMIGA.

Even if your knowledge of the AMIGA is limited, *MacLibrary* can ease your job of implementing screens, windows and gadgets by utilizing the functions, examples and sample programs included with the package.

Other *MacLibrary* routines are functionally compatible with the most widely used Apple® Macintosh™ Quickdraw Routines™, Standard File Package and Toolbox Utility Routines enabling you to rapidly convert your Macintosh programs to run on the AMIGA.

Panel™ ***$195.00***

Panel will help you write your screen programs and layer your screen designs with up to ten overlapping images. *Panel's* screen layouts can be assigned to individual windows and may be dynamically loaded from files or compiled into a program. *Panel* will output C source for including in your applications. A monitor and keyboard utility is also included to allow you to customize your applications for other systems.

With Lattice products you get *Lattice Service* including telephone support, notice of new products and enhancements and a 30-day money-back guarantee. Corporate license agreements available.

Lattice

Lattice, Incorporated
Post Office Box 3072
Glen Ellyn, Illinois 60138
(312) 858-7950 TWX 910-291-2190

INTERNATIONAL SALES OFFICES:
Benelux: Ines Datacom (32) 27205161 England: Roundhill. (0672)54675
Japan: Lifeboat Inc. (03) 293-4711 France: SFL (1) 46-66-11-55
Germany: (49) 7841/4500