THE MICROCOMPUTER MAGAZINE

commodore

FEBRUARY 1982
$2.50

VIC-20

VIC AVENGER IS HERE!

"LEGAL TIME ACCOUNTING PLUS WORD PROCESSING FOR UNDER $6,500. FROM COMMODORE."

—WILLIAM SHATNER

"Before we had our Commodore computer and LTA system, there was no way to keep complete track of our time. And if you can't account for time, you can't bill for it. This system keeps track of every minute spent and issues very detailed invoices. It also keeps us posted on outstanding accounts receivable before they become past due. The truth is, we've had our Commodore computer less than a year and it's almost paid for itself already."

—Don Nelson, Attorney-At-Law
Pasadena, California

No one has to tell you that the practice of law runs on a sea of paperwork. Sometimes it seems like you're drowning in it. Fortunately, a simple, inexpensive solution is at hand: a Commodore computer. Including disk drive, letter quality printer, legal time accounting and word processing programs.

For a modest investment, you get all the features of a sophisticated word processor and a versatile business computer that can help your office operate more efficiently than you ever thought possible.

Commodore's Legal Time Accounting (LTA)[1], for example, keeps track of all the services you perform for your clients and handles billing automatically. It enables you to see instantly who is performing what services for which client and how much time is spent on each activity. Aging analysis reports reflect outstanding receivables over a 90 day period.

The system is easy to use, even for those with no computer experience. Lawyers simply fill out a log sheet at the completion of each activity. A date is entered into the LTA system which stores the information for as many as 500 clients. The data can be cross-referenced to produce nine different reports, from daily summary journals to client billing statements.

And with our Wordcraft 80 word processing program, your Commodore computer is versatile enough to be used whenever you'd normally use a typewriter. For briefs. Memos. Correspondence. In seconds, you can delete, insert, rearrange paragraphs, revise, with no time wasted typing multiple drafts.

In short, a Commodore computer can free up your legal staff to do what it does best... practice law.

Your Commodore computer can be expanded to meet the needs of a growing office. And Commodore dealers throughout the country offer prompt local service.
Visit your Commodore dealer for a hands-on demonstration of the computer that does so much, so easily, at such a low cost.

1 Legal Time Accounting was created by Cimarron Corporation

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

☐ Please send me more information on the LTA System.

Name _____

Address _____

City _____ State _____

Zip _____

Phone _____

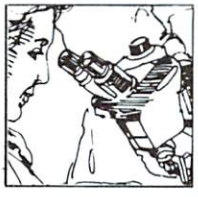
commodore COMPUTER

THE MICROCOMPUTER MAGAZINE

commodore

FEATURES

Commodore Computers Provide Market Research Firm With All The Right Answers 21



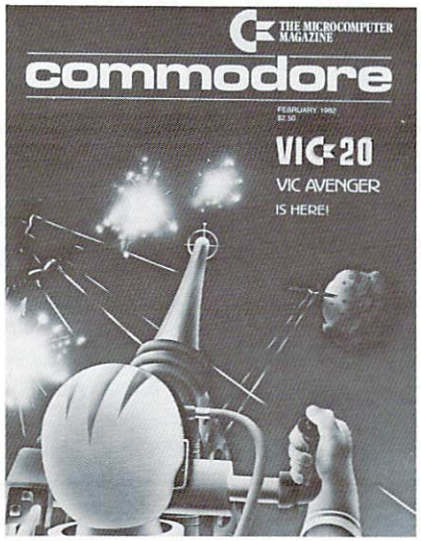
PETs Help Chemical Engineering Students to Control Experiments 27

Meet Musician George Kuetemeyer and His PET Vocalist, Drummer, and Bass Player 33

Take It from the VIC Magician. The VIC Is a Super Calculator! 39

Solution to a SuperPET Mystery: How to Access the RS-232 Serial Port Using the 6502 Processor . 58

DEPARTMENTS



commodore

Editorial Manager
David A. Kaminer
Creative Supervisor
Edythe Silverglide
Editor
Paul Fleming
Staff Editor
Jody Miller
Staff Writers
Rick Cotton
Andy Finkel
Paul Goheen
Neil Harris
Mike Heck
Pat McAllister
Dave Middleton
Sue Mittnacht
John O'Brien
Dave Scott
Michael Tomczyk
Contributing Writers
Jim Butterfield
Elizabeth Deal
D.G. Barker
Naomi Blackburn
Karl Hildon
Glen Pearce
Technical Staff
Rick Cotton
Paul Goheen
Neil Harris
Pat McAllister
Dave Middleton
Dave Scott
John Stockman
Michael Tomczyk
Circulation Manager
John O'Brien
Advertising Manager
Paul Fleming

Commodore Magazine is issued bi-monthly. Published by the Computer Systems Division, Commodore Business Machines, Inc., 681 Moore Rd., King of Prussia, PA 19406 (215) 337-7100. The U.S. Subscriber rate: $15.00 per year. In Canada and Mexico the Subscriber rate is $25.00. No material may be reprinted without permission. Volume 3, Number 1.

Q&A HOTLINE

Q. I have a PET 2001, originally 8K and still original ROM. I have installed Skyles 24K RAM board to bring RAM to 32K. For some time, I have wanted to upgrade the original ROM to "new" ROM, but without losing the rather extensive library of programs recorded on cassette tape in old ROM. BASIC SWITCH, from Applied Micro Systems of Mishawaka, Indiana, seemed to be my answer. As I understood BASIC SWITCH would allow me to switch back and forth from old ROM to new ROM. Eventually I would be able to convert my library of tapes to the more efficient ROM. I have tried unsuccessfully to contact Applied Micro Systems. Can you provide me with the name(s) of any computer product which fits my requirement. I am interested in a "switch-type" product, rather than one that needs soldering.

H. Stark
Coronado, California

A. *We tried to contact Applied Micro Systems and learned that they have been bought out by Competitive Software. We spoke to this Michigan-based company, who informed us that they were gearing up for production with deliveries to begin as early as February, 1982. They can be contacted at the following address:*

Competitive Software
21650 Maple Glen Drive
Edwardsburg, Michigan 49112
(616) 699-7115

Another excellent ROM switch is manufactured by Batteries Included in Canada. Their unit is called the SWARM-100 and allows for software selection of utility ROMs as well. Their address is:

Batteries Included
Village by the Grange
71 McCaul Street
Toronto, Ontario
Canada M5T 2X1
(416) 596-1405

Neither of these units mentioned require any soldering. Incidentally, Commodore's 64K add-on memory board for the 8032 comes with a disk which allows you to soft-load different versions of Commodore BASIC to emulate older machines.

Q. Our company has a 2001 series microcomputer, recently retrofitted to BASIC 4.0. On the carriage return, the cursor sometimes does not return to the normal position (the first column of the next line), but stays on the same line, jumping a few spaces. This problem was not noticed before the update was completed.

A. Jafri
Hyatsville, Maryland

A. *The problem you are having with the cursor is caused by* not *executing a PRINT# command after a CMD command, but before a close command. An example of the correct sequence is:*

```
10 OPEN 1,4
20 CMD 1
30 PRINT # 1
40 CLOSE 1
```

If line 30 was not executed before CLOSE 1, then you would experience the difficulties you described.

Q. I have a 2022 printer and a 4022 printer hooked up to an 8032. On certain programs the 2022 printer jumps into lower case and the only way to resolve this problem seems to be to turn the printer off and back on again. Is there any way to fix this problem? My 4022 doesn't do this but I would like to make them both print bi-directionally if possible.

W. Betz
East Lansing, Michigan

A. *The problem you are having in your 2022 printer is caused by a small bug in one of the older printer ROMs. (Part number 901472-03 or*

Continued on page 4.

IF YOU'RE WAITING FOR THE PRICE OF WORD PROCESSORS TO FALL WITHIN REASON,

Everyone expected it would happen sooner or later...with WordPro PLUS™ it already has! Now all the marvelous benefits of expensive and advanced word processing systems are available on Commodore computers, America's largest selling computer line. WordPro PLUS, when combined with the new 80 column CBM 8032, creates a word processing system comparable to virtually any other top quality word processor available—but at savings of thousands of dollars!

New, low cost computer technology is now available at a fraction of what you would expect to pay. This technology allowed Commodore to introduce the new and revolutionary CBM 8032 Computer.

WordPro PLUS turns this new CBM 8032 Computer into a sophisticated, time saving word processing tool. With WordPro PLUS, documents are displayed on the computer's screen. Editing and last minute revisions are simple and easy. No more lengthy re-typing sessions. Letters and documents are easily re-called from memory storage for editing or printing with final drafts printed perfectly at over five hundred words per minute!

Our nationwide team of professional dealers will show you how your office will benefit by using WordPro PLUS. At a price far less than you realize.

Invest in your office's future...
Invest in WordPro PLUS...
Call us today for the name of the WordPro PLUS dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
(617) 444-5224
TELEX: 95 1579

Professional Software, Inc. WordPro was designed and written by Steve Punter.

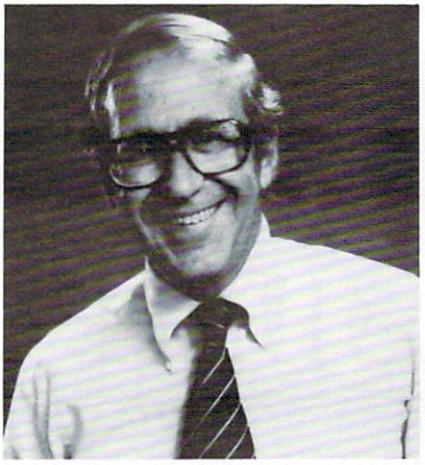
Q&A HOTLINE

901472-04.) There is now available a new printer ROM which is designed to correct this and other problems with the 2022. The new part is 901472-07.

Although it isn't possible for the 2022 to print bi-directionally, this can be accomplished on the 4022 printer by replacing the old ROM (#901490-01) with the bi-directional ROM (#901631-02). Contact your dealer for details. ■

Just can't get the answer to your questions on Commodore Equipment/ Applications???
Write: **HOTLINE**
c/o Commodore Magazine
681 Moore Rd.
King of Prussia, PA 19406

Message from the President



In 1981, Commodore made history by introducing the world's first full-featured color computer -- the VIC 20. For an industry that was virtually unheard of just five or six years ago, this low-cost home computer was, in itself, a trendsetting breakthrough. Now, just a year after VIC's entrance into the market, Commodore has set another new standard in this young industry. With the recent announcement of two new products, Commodore will offer the widest range of home/personal computers ever available from a single company.

These two additions to the growing line of the company's personal computers -- the COMMODORE ULTIMAX and the COMMODORE-64 -- continue to assert the Commodore policy of selling technically superior products at low prices. This price/performance ratio is obviously evident in the COMMODORE ULTIMAX, which will retail for only $149.95. Like the Atari® VCS and Intellivision®, this product will play exciting video games. But, unlike its competitors, the COMMODORE ULTIMAX offers REAL computing power.

Amazing graphics offer twice the resolution of any of the leading game machines on the market today. In fact, the resolution on the ULTIMAX is equal to that of the more advanced Apple 2™. The reason for this is simple: we produce sophisticated computers that can also be used, quite effectively, to play games. Along with impressive graphics, sound generation is also a standard feature of both of Commodore's new home computers.

While boasting all the major features of the COMMODORE ULTIMAX, the COMMODORE-64 can also lay claim to other capabilities, including the availability of advanced computer languages. At less than $600, the COMMODORE-64 offers a significant breakthrough in price, selling at one-half the cost of competitive computers. And its 64K memory capacity is greater than many of these more expensive products.

These new home computers also reflect Commodore's consistent dedication to using our unique capability of vertical integration. Both the sound generator and phenomenal resolution display were designed at MOS Technology, Commodore's semiconductor subsidiary. The dedication and research that wrought these amazing achievements are the normal method of operation at Commodore, and will surely result in more technological and price breakthroughs in the months and years to come.

The computer industry in general has come a long way since the first electronic computer. And the development of microcomputers has made this evolution even more amazing. Today, desk-top microcomputers offer more computing capacity than their crude ancestors that filled an entire room. They operate 20 times faster and feature larger memories. And these "computers on a chip" consume 1/3 the power of the smallest night light, occupy 1/30,000th the space and cost 1/10,000th as much.

Perhaps even more astounding is that we are just beginning to realize the power and cost-effectiveness of the microcomputer. And Commodore -- deeply committed to this computer revolution -- is also devoted to making our microcomputers affordable to as many people as possible. The COMMODORE ULTIMAX and the COMMODORE-64 are just two examples of "real" computing power at prices so low they seem "unreal." Commodore is the only company committed to utilizing the latest technology in microprocessor design, and producing state-of-the-art products.

Jim Finke

James Finke
President

Editor's Notes

In the December issue, some of the improvements in our magazine during 1981 were reviewed, and our commitment to an even better publication in 1982 was discussed as well. Spurred by the fantastic growth of the home computing industry, Commodore is not wasting time in pursuit of that goal. But, for once, the next major change will not occur within these pages. Instead, it will involve the creation of an entirely new publication—one that will cater exclusively to the home computer enthusiast.

In the early Spring of 1982, all Commodore Magazine subscribers will receive their first issue of Commodore's yet-unnamed home computing magazine. Like our current publication, the new magazine will offer various features and application stories, beginners tips, new product announcements, technical information, and additional content limited only by the needs of the reader.

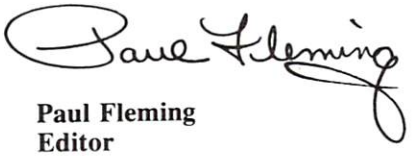
What will be the focus of this new magazine? Obviously, the popular VIC 20 will receive much of the early attention of our readers. But joining the VIC 20 (see Commodore News) in 1982 will be two new programmable/color low-priced home computers-the "COMMODORE ULTIMAX" and the "COMMODORE-64."

These two products, along with the VIC 20, will receive the kind of attention they deserve—and that only a separate publication could provide. Equally important is the amount of extra space we will be able to devote to our PET, CBM, and SuperPET products in Commodore Magazine, which we will continue to expand and improve.

However, for those of you who subscribe to Commodore Magazine solely for the VIC information, the format of this magazine will not change for at least one year. As your subscription to the Commodore Magazine expires, we will give you the opportunity to renew or change your subscription to the new publication.

Although price and frequency of publication have not been finalized, the initial issues of the new magazine will probably be published on a quarterly basis. Rest assured that all our readers will be kept up-to-date on the progress of the new publication and the subsequent policy regarding subscriptions.

The potential of the home computing industry is staggering. Along with the new and exciting personal computers offered by Commodore in 1982, we will be ready with an equally exciting publication to make home computing even more fun and educational.



Paul Fleming
Editor

A Note to Subscribers

Due to the growing demand for back-issues of our publication, some magazines are either very low or completely depleted. Those issues that are no longer available are volumes 4/5 (double issue), 6, and 7 of the old User Club Newsletter. All back issues of Interface and Commodore Magazine are still available.

Also, please notice that the mailing label affixed to this magazine contains your name and address, as well as a code number. This number indicates the month and year your subscription expires. For example, if the label reads "6/82," this means that in June of 1982 your subsription will expire.

If you have any problems with your subscription, please feel free to contact me at (215) 337-7100 or write in care of this magazine. If you correspond by letter, please include your phone number so you can be contacted immediately. ■

John O'Brien
Circulation Manager

Commodore Customer Support Team

In an effort to more effectively meet the needs of our customers, Commodore has created a Customer Support Team to assume the responsibilties previously handled by the Commodore HOTLINE.

As this magazine is being printed, the following changes and improvements are in effect:

- The Commodore HOTLINE number (800-523-5622) will be discontinued.
- The new Customer Support telephone number is (215) 337-1603. An increased staff of Customer Support personnel will answer calls from 9:00 a.m. to 5:00 p.m. EST, Monday through Friday.
- Written inquiries can be directed to:

  Customer Support Team
  Commodore Business Machines
  681 Moore Road
  King of Prussia, PA 19406

  Depending upon the complexity of the written inquiry, responses will be made either by telephone or through the mail. Noteworthy requests may even be reprinted in Commodore Magazine.
- The Customer Support staff has been organized to respond to the following types of inquiries: (1) general information regarding product availability, requests for literature, etc; (2) semi-technical information such as instructions on loading a program, explaining the keyboard, etc.; and (3) technical questions which, if Customer Support cannot answer, will be appropriately referred.
- Customer Support will also provide valuable feedback to our regional offices, customers, dealers, and headquarters. This feedback will take the form of sales inquiries, customer leads, dealer or customer grievances, and technical updates where applicable.
- When a call is placed, the caller will be handled professionally and courteously by one of our qualified support members. If all lines are busy, the call will be placed in a queue and a recorded message will inform the caller to wait for the next available open line.

These changes have been constructively planned in the very best interests of our customers. The HOTLINE was not fulfilling the role for which it was created. Persons were placing calls—often two or three times daily—to ask questions that could have been answered simply by referring to readily available sources, including product documentation, Commodore Magazine, area user clubs and highly competent dealers. Consequently, many callers with real and immediate problems were unable to get through with well-thought-out questions.

Hopefully, with this new approach, users will consider their problems more thoughtfully prior to placing a call. In turn, our Customer Support team will have more valuable time to spend with those who require our immediate response.

Any questions regarding customer support may be directed to:

Customer Support Manager
681 Moore Road
King of Prussia, PA 19406
(215) 337-7100 ■

Customer Support to the Rescue

One of the ongoing features in this magazine will be a list of the most frequent questions encountered by our Customer Support Team. For this issue, ten questions concerning the VIC 20 are discussed.

Frequent Questions About the VIC 20 Personal Computer

1. Q: How do I hook up a MODEM to the VIC?

A: Two choices are available. One way is to purchase the VIC RS-232 Terminal cartridge and use existing RS-232 acoustic MODEMs. If, however, you do not have an acoustic MODEM, your best bet is to get the VIC MODEM, initial deliveries are sceduled for March, at your Commodore dealer. With this inexpensive cartridge you simply insert the handset cord of your modular phone directly into the MODEM, run the software driver, and you are set for the world of computer telecommunications!

2. Q: Will more software be available for the VIC?

A: Commodore's VIC Product Development Group is currently working on several new and exciting arcade-style cartridge games as well as helpful application software.

3. Q: How do I get VIC schematics? Memory map?

A: Schematics, memory map, and other technical information for the VIC 20 are included in the Programmer's Reference Guide which will be delivered to Commodore dealers in February.

4. Q: How do I get on the VIC 20 mailing list?

A: Simply send in your completed VIC Warranty Registration Card and you will be put on our list. However, one of the best ways to keep informed is to subscribe to Commodore Magazine.

5. Q: Is the PET/CBM software adaptable for the VIC?

A: Generally speaking, much PET/CBM software can be adapted to the VIC if there is sufficient memory. Programs most easily adapted are those which are in BASIC and contain no PEEKs or POKEs. Because memory configurations vary from computer to computer, machine code programs and BASIC programs with PEEKs-/POKEs are often difficult to adapt. Converting programs not only gives you more programs in your VIC program library, but also helps you learn more about programming and program structure.

6. Q: What does the Superexpander do? What is the pixel matrix size?

A: The Superexpander is a cartridge which not only expands the VICs memory by 3 kilobytes but also gives you high resolution graphics plotting, color, print, and sound commands. All commands may be typed as new BASIC commands or accessed by hitting one of the VIC's special function keys.

Superexpander has 1024 x 1024 dot screen plotting. The resolution of the VIC screen is 176 x 184 pixels. The superexpander maps down to the actual resolution of the screen.

7. Q: Can we add more than 32K RAM?

A: The 6502 microprocessor, which is the heart of the VIC 20, can only address about 65 kilobytes of memory. Much of the memory in the VIC is ROM, which is already allocated to system routines. When we subtract the used ROM memory from the possible 65K of memory this leaves room for approximately 32 kilobytes of RAM.

8. Q: Are joysticks needed with game cartridges?

A: With the current VIC cartridge games, joysticks are not necessary. They are optional on the following: VIC Avengers, Superslot, VIC Super Alien, and Radar Ratrace.

9. Q: Why do I have problems loading programs from cassette tapes?

A: Radiation from the television screen can sometimes interfere with the loading of a program on tape from any cassete recorder. To alleviate this problem, move the recorder as far as possible from the televsion. Because new cassette tapes are often tightly wound, we suggest that you play the tape a few times without loading. This will loosen the tape and facilitate loading.

10. Q: Do the various application software cartridges reduce the amount of RAM available?

A: The Programmers Aid and VICMON Machine Language Monitor cartridges do not reduce the amount of RAM accessible to BASIC. The VIC 20 Super Expander *adds* to Random Access Memory (3 kilobytes). ■

— Eric Cotton & Sue Mittnacht

Welcome Aboard!

Congratulations to the following groups, who have joined the lengthening ranks of dealers selling Commodore computer products...

MPX Assoc.
3192 A. Lewiston
Berkely, CA 94705
415-642-5452

Properties Unlimited Inc.
548 Rose Lane
Paso Robles, CA 93446
805-238-7859

Ternes Office & Stationary
110 Liberty St.
Petaluma, CA 94952
707-762-9403

Cash Register Systems
313 Kamakee St.
Honolulu, HI 96814
808-533-7806

Western Kansas Computer Serv.
2606 N. Fleming
Garden City, KA 67846
316-276-8326

Delaware Valley Computer Serv.
212 Mimosa Dr.
Cherry Hill, NJ 08003
609-424-2875

Electroscience
7816 Ridge Rd.
Brockport, NY 14420
716-637-5102

Personal Computers Inc.
95 Niagara Falls Blvd.
Buffalo, NY 14214
716-832-8800

Computer Home
431 East Ave. C, Box 1966
San Angelo, TX 76903
915-653-7488

Southwest Micro Bus. Computer
8310 S. W. Freeway
Houston, TX 77074
713-771-5265

A-1 Services
8300 Gage Blvd. #221
Kennewick, WA 99336
509-783-4980

Allied Business Systems
2320 Broadway
Oakland, CA 94612
415-444-6383

Huntington Computing
1945 S. Dairy Ave.
Corcoran, CA 93212
209-992-4481

MicroPacific Computer Center
5148 N. Palm
Fresno, CA 93704
209-229-0101

Conserve
106 E. 2nd Ave.
Grandview, WA 98930
509-882-2947

Electronic Super Mart
17370 South Center Pkwy
Seattle, WA 98188
206-575-1093

B.B.M.
Oakland Center 9030 Route 108
Columbia, MD 21045
301-997-5500

Best Business Equipment
271 Lincoln St.
Worchest er, MA 01605
617-755-1077

Portsmouth Computer Center
31 Raynes Ave.
Portsmouth, NH 03801
603-431-7438

Wolff Office Equipment
1841 Broadway
New York, NY 10023
212-581-9080

American Computer Co.
1004 8th Ave.
Nashville, TN 37203
615-242-2592

USERS CLUBS: Sound Off!

We're continuing to compile a list of all Commodore Users clubs throughout the country. To date, our list includes the names mentioned on this page. If you'd like to add your name to the rolls, please send your club's name, address, and other pertinent information to:

Commodore Users Clubs
c/o Editor
Commodore Magazine
681 Moore Road
King of Prussia, PA 19406

And remember, once our list is comprehensive enough, we will begin forwarding valuable information to clubs on a regular basis, including hardware and software updates, technical bulletins, new product announcements, and troubleshooting tips.

CALIFORNIA

Lawrence Hall of Science
UC Berkeley
Computer Project, Room 254
Berkely, CA 94720
(415) 642-3598

Downey-Bellflower Users Group
c/o Robert Johnson
14944 Bayou Avenue
Bellflower, CA 90706

Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA
(213) 849-4094
1st Wed. 6pm

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

BAMBUG
1450 53rd Street
Emeryville, CA
(415) 523-7396

North Orange County Computer Club
3030 Topaz, Apt. A
Fullerton, CA 92361
Dave Smith

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, secretary

PALS (PETs Around Livermore Society)
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
314 10th Avenue
Oakland, CA
(415) 451-6364
Every 2nd & 4th Thurs.

Sacramento PET Workshop
PO Box 28314
Sacramento, CA
(916) 445-7926
Every 3rd Thurs-7:30 pm

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 am-4 pm

Walnut Creek PET Users Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

FLORIDA

Jacksonville Area PET Society
401 Monument Road, #177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee Avenue
Chicago, IL 60647

Central Illinois PET Owners
Rick Goldsmith
2730 Townway Road
#E-54
Danville, IL 61832

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt, president

INDIANA

PET Users
Jerry Brinson
PO Box 36014
Indianapolis, IN 46236
(317) 898-3604

GHS Computer Club
c/o Grangevile High School
910 S D St.
Grangeville, ID 83530
(208) 983-0580
Contact: Don Kissinger

IOWA

PET Users Group
c/o Don Vorhies
1321 42 St. SE.
Cedar Rapids, IA 52403

MARYLAND

Assoc. of Personal Computer Users
5014 Rodman Road
Bethesda, MD 20016

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

PET User Group
Peter Oakes
2235 Lakeshore Drive
Muskegon, MI 49441

Toledo PETS
734 Donna Drive
Temperance, MI 48182
Contact: Gerald Carter, president

MINNESOTA

Twin Cities
John Fung
Twin Cities, MN
(612) 376-5465

MISSOURI

St. Louis Club
Mary Perkinson
46 Westwood Court
St. Louis, MO 63131
(314) 432-5225

NEVADA

Las Vegas PET Users
4884 Iron Avenue
Las Vegas, NV 89110

NEW JERSEY

Amateur Computer Group of New Jersey
John Loofbourrow
UCTI, 1776 Raritan Road
Scotch Plains, NJ 07076
(201) 233-7068

Amateur Computer Group 18 Alpine Drive
Wayne, NJ 07470

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

NEW HAMPSHIRE

Northern New England Computer Society
PO Box 69
Berlin, NH 03570

NEW YORK

Capital District PET Users
Ben Green
Albany area, NY
(518) 370-1820

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club of Westchester
Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

PET User Group
Westchester, NY
(914) 428-7872
Every 2nd Tuesday

PET User Group
c/o Meyer
35 Barker Avenue
White Plains, NY 10610

OHIO

Dayton Area PET User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, president
(513) 848-2065
J. Watson, secretary
(513) 372-2052

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

PENNSYLVANIA

PET User Group
Gene Beals
PO. Box 371
Montgomeryville, PA 18936

PACS PET Users Group
20th & Olney Streets
Philadelphia, PA

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

TENNESSEE

River City Computer Hobbyists
Memphis, TN
1st Mon. at Main Library

TEXAS

SCOPE
1020 Summit Circle
Carrolton, TX 75006

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
PO Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M Micro-computer Club
Texas A & M, TX

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore User's Club
742 Taylor Avenue
Ogden, Utah 84404
Contact:
Todd Woods Kap, president;
David J. Shreeve, vice president

VIRGINIA

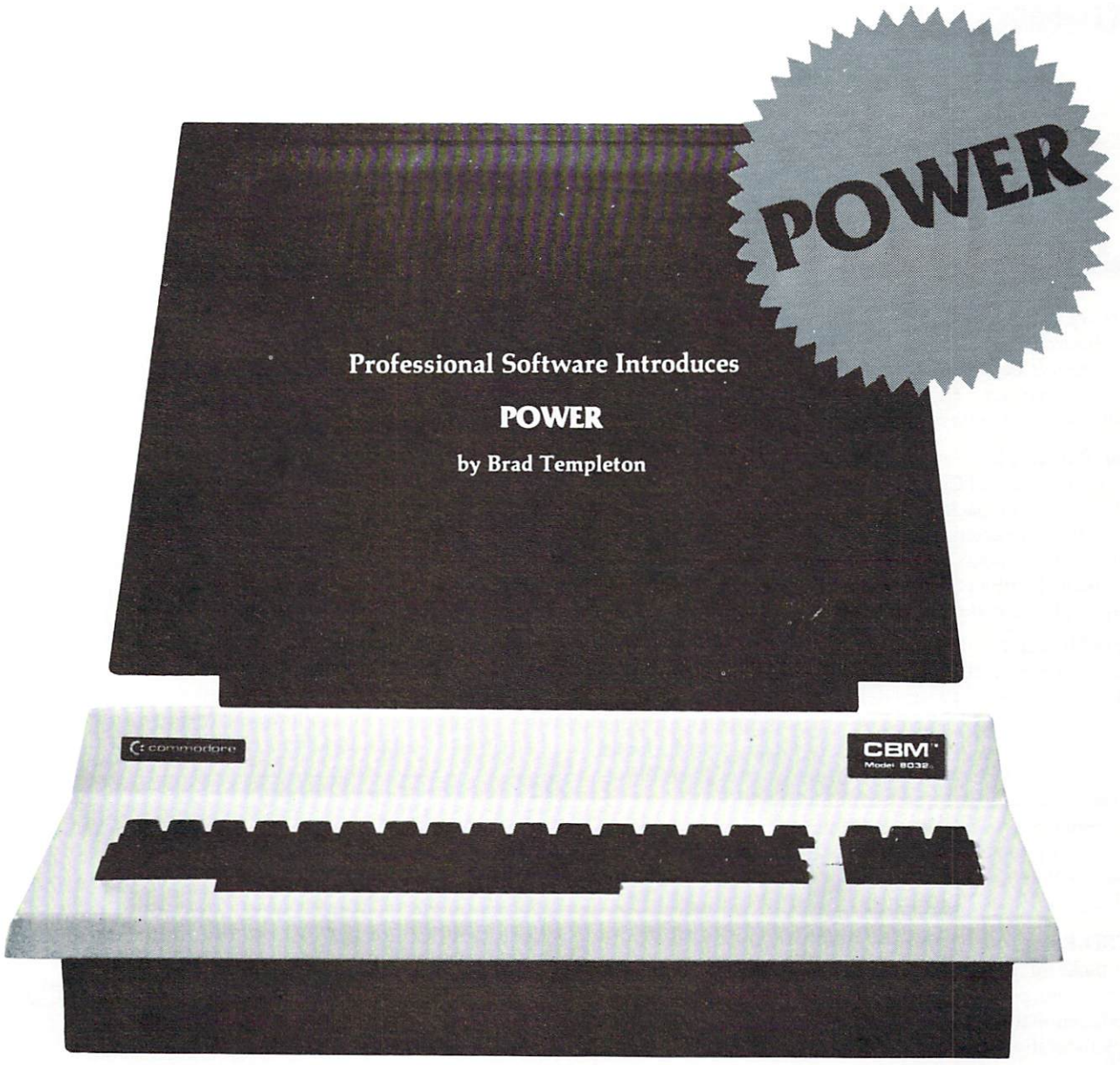
Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(703) 860-9116

WASHINGTON

Northwest PET User Group
PO Box 482
Vashon, WA 98070

WISCONSIN

Sewpus
c/o Theodore J. Polozynski
PO Box 21851
Milwaukee, WI 53221



ADD POWER TO YOUR COMMODORE COMPUTER

$89.95

POWER produces a dramatic improvement in the ease of editing BASIC on Commodore's computers. POWER is a programmer's utility package (in a 4K ROM) that contains a series of new commands and utilities which are added to the Screen Editor and the BASIC Interpreter. Designed for the CBM BASIC user, POWER contains special editing, programming, and software debugging tools not found in any other microcomputer BASIC. POWER is easy to use and is sold complete with a full operator's manual written by Jim Butterfield.

POWER's special keyboard 'instant action' features and additional commands make up for, and go beyond the limitations of CBM BASIC. The added features include auto line numbering, tracing, single stepping through programs, line renumbering, and definition of keys as BASIC keywords. POWER even includes new "stick-on" keycap labels. The cursor movement keys are enhanced by the addition of auto-repeat and text searching functions are added to help ease program modification. Cursor UP and cursor DOWN produce **previous** and next lines of source code. COMPLETE BASIC program listings in memory can be displayed on the screen and scrolled in either direction. POWER is a must for every serious CBM user.

Call us today, for the name of the Professional Software dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
Tel: (617) 444-5224 Telex #951579

™POWER is a registered trademark of Professional Software, Inc.

Software for the 64K Memory Expansion Board

Since we announced the availability of the Memory Expansion Board (October, 1981), a great deal of application software has been designed and introduced to complement the new product. Here's a list of these new software developments. Contact your local dealer for more details.

8096 Wordcraft
This new version of Commodore's own "industry standard" word processing package has been enhanced so that it now has 24,000 characters of workspace available per document. Other enhancements include the provision of communications between PETs of documents of up to 12-15 pages long, using sophisticated error recovery techniques.

MicroModeller
MicroModeller is a financial modeling system, designed to make it easy to specify a model, put in data, generate reports or graphics, and ask 'WHAT IF' questions. MicroModeller has been designed to be used at two levels: 1) by the model developer who can afford to spend time learning the language in return for access to powerful features which speed the development process; and 2) by the end user who doesn't have time or possibly the background to learn the nitty gritty details of MicroModeller. Sold by Intelligence (UK) Limited, 30 Lingfield Road, London, SW19 4PU.

UCSD Pascal
For those of you that have waited for a truly 'structured' programming language for serious application development or educational purposes, UCSD Pascal is here. This is a full implementation of all the latest enhancements and extensions of the language. It will be sold as a developmental system with full compilation support, and as a 'run time' only system to execute those applications that are designed using UCSD Pascal as a base. Sold by Commodore.

Silicon Office
Silicon Office is one of the world's most advanced microcomputer systems. It is designed specifically to allow you, the user, freedom to use your computer as a professional tool. It greatly simplifies computer instructions, allowing you to "drive" your computer in any way you choose, without the need for expert programming knowledge. This product includes all of the functions of an integrated DBMS and a very good wordprocessing system combined into one. The system also supports communications between computer systems in the form of file transfer, direct communication, and data sharing. With this tool, applications from a very simple mailing list to a complete Order Entry and Invoicing system may be generated with little effort. Sold by Bristol Software Factory, Kingsons House, Grove Avenue, Queen Square, Bristol, UK BS1 4QX.

EASY
EASY is a software system designed to aid you in the bookkeeping and accounting needed to make your household or small business run as efficiently as possible. It is an easy-to-use package which allows you to record transactions without having to know anything about computer programming and to have only minimal knowledge of accounting. A very good 'starter' package that can grow with you and your business. Sold by Denver Software Co., 14100 East Jewell Ave., Suite 15, Aurora, CO 80012

WordPro 5+
This version of the ever popular WordPro wordprocessing software series has been designed to utilize the additional memory made available with the 64K Memory Expansion Board. It now has the capability to contain up to five 182 line areas of text in memory at the same time. Other functional enhancements are implemented as well. A good product is now even better! Sold by Professional Software, Inc., 166 Crescent Road, Needham, MA 02194.

The Commodore 64K Memory Expansion Board
This product is designed to work with the CBM 8032 or 4032 and will add 64K bytes of additional RAM, providing for a total of 96K RAM of program area. The board is a "plug-in" module to the main logic board of these machines and is attached via four mounting brackets and three cables (2 power cables, and 1 interface cable). A diskette containing programs for controlling the expansion memory is also supplied with the board.

Features
- A loader program which allows the user to load in and run one of the three ROM versions of CBM BASIC.
- A machine language monitor which provides access add-on memory.
- A set of added BASIC commands for use of the add-on memory from BASIC programs.

Advantages
- Ability to keep an application's "menu" program and several frequently-used modules resident in memory so that program chaining is much faster.
- Through the organization of source code into small modules, a combined program and data area of 70K can be achieved when programming is in BASIC.
- Assembler language applications can be up to 96K bytes.
- Ability to load in large amount of data in the extra memory and access this data via the additional BASIC commands. ■

Professional Business Software

For The Commodore 8000 Series Computer System

CMS GENERAL ACCOUNTING SYSTEM II:

A fully interactive General Accounting System designed especially for the first time user. All input requests are fully prompted with complete verification of input data. Most reports may be printed either to the screen or the printer and started or stopped at any point. The user is led completely through each function by a series of highlighted prompts fully explaining the required input at each point. A professionally written instruction manual is included which shows sample reports generated by the system and further explains each step and prompt as it is encountered by the user. These user prompts, together with the detailed step by step manual, make it virtually impossible for the user to accidentally crash the program or to get lost in the program and be unable to proceed or backup. Some of the many features of each of the four major accounting functions is shown below.

GENERAL LEDGER:

Up to a 1000 accounts on the Chart of Accounts. Fully departmentalized up to nine departments. Cash Disbursements and Cash Receipts Journal as well as a General Journal for ease of data entry. Maintains account balances for Present Month, Quarter to Date, and Year To Date. User customized financial statements. Accepts postings from Accounts Receivable, Accounts Payable, Payroll, or other programs.

ACCOUNTS RECEIVABLE:

Prints Invoices and Monthly Statements. The finance charge rate and period may be set by the user. Full invoice aging reports with aging breaks set by the user. During invoice data entry a copy of the Invoice is displayed on the screen and the information is typed in exactly as if the Invoice was in a typewriter. Accomodates full or partial invoice payments. Provides for Credit and Debit Memos as well as Invoices. Invoice File capacity is 2000 minus the number of customers multiplied by 1.4. Five hundred customers will allow room for 2100 invoices. Invoices may be distributed among up to nine different General Ledger accounts with automatic updating to the General Ledger.

ACCOUNTS PAYABLE:

Prints Accounts Payable checks with full check voucher detail for each Invoice paid. Prints detailed check register. Automatic application of Credit Memos. Complete invoice aging reports with aging breaks set by the user. Invoice File capacity is 2000 minus the number of vendors multiplied by two. Invoices may be distributed among up to nine different General Ledger accounts with automatic updating to the General Ledger Account File.

PAYROLL:

Maintains Monthly, Quarterly, and Yearly totals for each of up to 350 employees. Prints Payroll checks with full deduction and pay detail. Accomodates Weekly, Bi-weekly, Semi-Monthly, and Monthly employees. Pays regular, overtime, holiday, and piece work hours. Up to eight miscellaneous deductions or payments per employee. Prints Payroll Journal, Payroll Check Register, and an Absentee Report as well as 941 information and W2 forms. Automatic updating to the General Ledger.

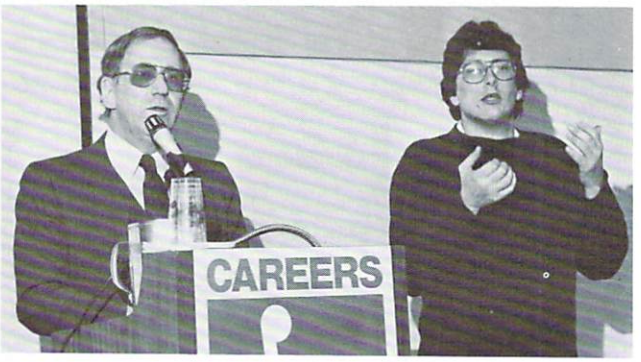
See Your Nearest Commodore Dealer For A Demonstration

CMS Software Systems, Inc. 2204 Camp David, Mesquite, TX 75149 214-285-3581

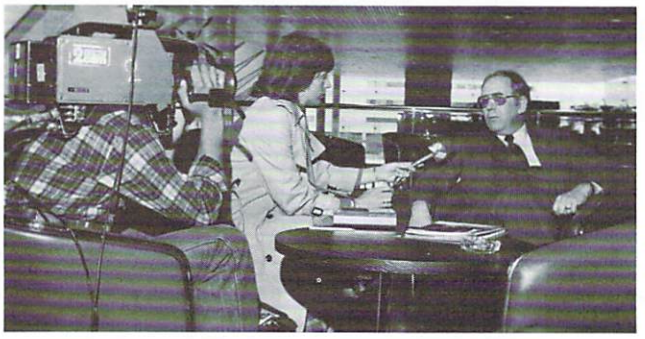
Commodore Lauded for Contributions to Baltimore Symposium

Commodore's "Careers for the Disabled Symposium" in Baltimore was highlighted by a day-long workshop on "data processing careers for the handicapped" presented by this distinguished "faculty." Standing, from left, are, Jack Reid, assistant manager—Human Resources, Manufacturers Hanover Trust Company, New York City, Vic Knorr, industry specialist, Electronic Industry Foundation, Washington, D.C., Dr. Bruce Downing, director of training, Commodore Business Machines, Inc., Valley Forge, PA, and Art Murphy, program instructor, and Joy Kniskern, program manager, Computer Programming Project for Severely Handicapped Persons, a Goodwill Industries program in Atlanta. Seated in front is Gene Spalding, president of Computer Systems, Inc., of Columbus, GA. Several hundred handicapped persons from throughout the United States attended the three-day symposium and heard from more than 40 experts on career opportunities for the disabled.

"Data Processing Careers for the Handicapped" was the subject of a day-long workshop held during the Commodore Business Machines sponsored "Careers for the Disabled Symposium." Here, Dr. Bruce Downing, director of training for Commodore, addresses the workshop while a "signer," at left, translates his talk into "sign language" for the hearing impaired. Several hundred handicapped persons from throughout the United States attended the three-day symposium.



Commodore president Jim Finke, left, speaks to the audience at the Careers for the Disabled Symposium in Baltimore in December, as a "signer" translates his words into "sign language" for the hearing impaired.



There was media attention aplenty focused on the Commodore-sponsored Careers for the Disabled Symposium in Baltimore in December. Here, Jim Finke, president of Commodore, is interviewed by Fran Franshel of WMAR-TV.

Gene Spalding, president of Computers Systems, Inc., a Commodore dealership in Columbus, GA, was one of the featured speakers at the Careers for the Disabled Symposium in Baltimore. Gene, a quadriplegic, spoke to the attendees about the numerous opportunities for the handicapped concerning microcomputers.

Proclamation

BY

MAYOR WILLIAM DONALD SCHAEFER

DESIGNATING DECEMBER 4,5, AND 6, 1981

AS

"CAREERS SYMPOSIUM DAYS" IN BALTIMORE

WHEREAS, Commodore Business Machines, Inc., in association with Careers for the Disabled, Inc., is sponsoring this Careers Symposium in Baltimore City; and

WHEREAS, although the employment of disabled persons has increased significantly over the years, our Nation's disabled still lead statistically in unemployment; and

WHEREAS, it should be recognized that ability counts when performing a job, not disability, and that many barriers, both architectural and attitudinal still exist; and

WHEREAS, disabled workers are entitled to recieve the same consideration as other applicants when being considered for a position; and

WHEREAS, job opportunities must be made available to all disabled persons in order to allow them the opportunity to share in our Nation's growth and lead independent lives.

NOW, THEREFORE, I, WILLIAM DONALD SCHAEFER, MAYOR OF THE CITY OF BALTIMORE, do hereby proclaim DECEMBER 4,5,AND 6, 1981 as "CAREERS SYMPOSIUM DAYS" IN BALTIMORE, and I urge all citizens to recognize the advantages of hiring the disabled.

IN WITNESS WHEREOF, I have hereunto set my hand and caused the Great Seal of the City of Baltimore, to be affixed this fourth day of December, in the year of Our Lord, one thousand nine hundred and eighty-one.

William Donald Schaefer
Mayor

CES Show Best Ever for Commodore!

Home Computer Interest at the annual Winter Consumer Electronics Show in Las Vegas was at an all-time high this year, especially at the booth showing Commodore Computers. Computer dealers, mass merchandisers, and many others helped make the display of VIC 20 home computers and other new home computer products one of the most popular exhibits at the show. "There were some 70,000 people at CES," said Commodore's vice president-marketing Kit Spencer, "and it seemed as if they all visited us to learn about the VIC and our other revolutionary new home computer products."

The Consumer Electronics Show held January 8-11 in Las Vegas was Commodore's most successful U.S. trade convention ever. The event is a showcase for new products and this year Commodore dedicated more than 90% of the booth space to the VIC 20 . . . clearly one of the "hottest" products displayed at the show.

The booth was literally "swarming" with dealers, distributors, and buyer teams. Everyone wanted to know why Commodore had Bally arcade vending machines in the booth . . . the answer . . . Commodore's licensing agreement with Bally provides for the conversion of such top-selling vending machine games as GORF, OMEGA RACE and WIZARD OF WOR to plug-in cartridge for the VIC 20. These games should be available by early Spring.

Some of the cartridge software displayed at the show included SARGON II CHESS, RADAR RATRACE, VIC AVENGERS, JUPITER LANDER, ROAD RACE, SUPERSLOT, DRAW POKER, SUPER ALIEN and MOLE ATTACK. The first HOME CALCULATION six-pack of programs on tape was also shown, including a wordprocessing program which will retail for under $15.

A unique import from the United Kingdom is INTRODUCTION TO BASIC, Part I, a workbook with two tapes containing 17 programs which help new computerists teach themselves programming. The self-teaching book-and-tape set includes a plastic flowcharting stencil to help new programmers plan out their own programs before they start. More titles will be offered as part of Commodore's TEACH YOURSELF PROGRAMMING series of books and tapes.

Another popular item was VIC-MODEM . . . the lowest-priced telephone modem in the computer industry. The new modem helps confirm the VIC 20 as a "home appliance" rather than a hobbyist device. The modem allows VIC owners to obtain stock quotes, wireservice news articles, even leave electronic "mail" messages for other computer owners... simply by connecting their VIC to a telephone. The modem goes on sale in March. Commodore will soon unveil a special telecomputing "network" for Commodore computer owners, with availability scheduled for Spring of this year.

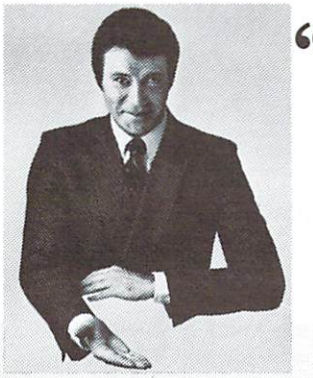
The "Talking VIC" was on exhibit, using VOTRAX's "Type N' Talk" voice synthesizer.

Even William Shatner, of STAR TREK fame, was there—on tape. Shatner, who is Commodore's spokesperson, will be appearing in a series of television advertisements featuring VIC 20, the "Wonder Computer of the 80's." The new TV commercials and a 10-minute videotape were shown, featuring Shatner explaining what the VIC 20 is and how easy it is to use.

Look for Commodore's exciting commercials during the Grammy Awards telecast, as well as major sporting events, including the Kentucky Derby and Indianapolis 500.

Retailers were especially impressed by Commodore's new VIC 20 in-store sales fixture. The new modular store unit includes a complete working VIC 20 system with peripherals and a full assortment of distinctively packaged software cartridges, tapes and books. The fixtures are already being installed in some of the largest mass merchandising chains in the country.

In the aftermath of the Consumer Electronics Show, Commodore has had to expand its sales administration staff to handle the huge influx of orders from mass merchandisers, distributors and retail stores. New VIC dealers include audio-video stores, toy stores, electronics boutiques, catalog stores, and many of the largest department stores in the country. Many are putting computer centers into their stores for the first time . . . noting that video games and home computers were among the few "hot items" which sold well this past Christmas, in spite of the economic recession. ■



"MEDICAL ACCOUNTING PLUS WORD PROCESSING FOR UNDER $6,500. FROM COMMODORE."

—WILLIAM SHATNER

The symptoms are common. Missing receipts. Overdue invoices. Neglected insurance forms. And, worst of all, a lot of precious time spent on paperwork that could otherwise be devoted to patient care.

The cure: A Commodore desktop computer. Including disk drive, letter quality printer, and complete medical accounting and word processing systems. For a modest investment, you get all the features of a sophisticated and versatile business computer that can do virtually all your paperwork in a fraction of the time it takes you now.

Commodore's Medical Accounting System (MAS)[1], for example, can provide you with a fast, flexible accounting and bookkeeping system that's as easy to use as it is cost effective. Automating your receivables, invoicing, aging of payables, and revenue analyses. MAS can also generate end-of-the-month "Superbills" as well as standard insurance and Medicare forms. And it gives you a thorough overview of your office activities through a series of reports ranging from diagnostics to referrals.

And with our word processing programs, your Commodore computer is versatile enough to be used whenever you'd normally use a typewriter. For memos. Reports. Correspondence. Proposals. In seconds, you can delete, insert, rearrange paragraphs, even revise as many times as necessary. With no time wasted typing multiple drafts.

If all that time saved on paperwork is used to take on additional patients, just think how quickly your Commodore computer will pay for itself, many times over.

Your Commodore computer can be expanded to meet the needs of a growing office. And Commodore dealers throughout the country offer prompt local service. Visit your Commodore dealer for a hands-on demonstration of the Commodore computer that does so much, so easily, at such a low cost.

1 Medical Accounting System was created by Cimarron Corp.

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

☐ Please send me more information on the MAS System.

Name _____

Address _____

City _____ State _____

Zip _____

Phone _____

commodore COMPUTER

COMMODORE NEWS

Expanded Memory for Program, Data Storage Offered by VIC 20's New Single Disk Drive

Expanded memory for storage programs and data is available to users of Commodore's VIC 20 home computer with the introduction of a new floppy disk unit.

The disk unit will store 170,000 information characters on standard 5¼-inch floppy diskettes. A built-in expansion port allows the disk unit to be attached to the VIC without additional interfacing or expense.

The VIC disk unit is an "intelligent" peripheral, which means no additional "user memory" is used up when the drive is used. It is compatible with Commodore's larger CBM 4040 dual drive and CBM 2031 single disk drive. And, as such, applications developed on Commodore PET® and CBM™ microcomputers can be easily transferred for use on the VIC.

Retail price of the VIC 1540 single disk drive is $595.00, and initial customer deliveries are planned for early spring of 1982. ■

New Home Calculation 'Six-Pack' Makes VIC 20 Home Computer Even More Valuable

The VIC Programmers Reference Guide, which provides complete information about the operation and programming of the VIC 20 home computer, is now available from Commodore.

The new programmers guide was compiled from the experience of Commodore's international programming staffs in more than half a dozen countries, and is designed for use by first-time computerists as well as experienced programmers.

To cover areas programmers are most interested in, the book is divided into four sections. The reference guide's complete "dictionary" includes BASIC commands as well as sample programs. The layman's overview to machine language programming gets you started writing machine code. The interface section shows how to expand the VIC 20 for telecommunications, joysticks, game paddles, and lightpens. Programming graphics and sound is covered in the final section.

Retail price of the VIC 20 Programmers Reference Guide is $16.95. ■

New Programmers Reference Guide Introduced for VIC 20 Home Computer Users

VIC 20 home computer users will find their full-color programmable computer more valuable than ever thanks to the introduction of the new Home Calculation "six-pack" specially tailored for American households.

Designed for use with Commodore's Datassette tape recorder, the package consists of six cassettes containing Personal Finance, VIC Typewriter Word Processing, Expenses, Loan and Mortgage, and Home Inventory programs. The cassettes sell separately for $14.95, while the entire "six-pack" is available for only $59.95.

Personal Finance is a two-tape set that allows the user to budget expenses. The VIC Typewriter is a "mini word processor" that utilizes the VIC 1515 Graphics Printer. The Expense Calendar program tracks important income and expense records, as well as pertinent personal dates, which can be recorded and displayed at any time.

The Loan and Morgage Calculator aids decision-making, allowing VIC users to calculate the effects of different interest rates. The program provides amortization schedules and can calculate variable rate mortgages. The Home Inventory program provides an effective method of listing all home belongings. Ideal for insurance needs, this program allows the user to categorize items by serial number or value. ■

Home Computers Will Become 'Fixtures' in Home Just Like Television Says Commodore's Spencer

"Home computers are not simply a fad. They will not go the way of hula hoops or simple home video games. Clearly, they are here to stay, and will become part of the American home in the 1980s much as television sets became a generation earlier, and the 'two-computer family' will become a reality soon."

So says Kit Spencer, vice president—marketing of Commodore's Computer Systems Division.

When Commodore pioneered the microcomputer industry in the mid-1970s with the introduction of the PET®," Spencer went on, "it marked the dawn of an entire new era that has become the 'era of the home computer'. No longer are people afraid of working with computers, and no longer are computers too costly for most people or families. Today, VIC 20 home computers can be had for less than the price of color TVs, and with them, users can reach well beyond previous limits for education, recreation, and honest-to-goodness data processing—and they can do so right in their own homes.

"Certainly video games helped pave the way for home computers," Spencer said. "Because people learned to hook up these toys and games to their own TVs, the way was cleared for computers to finally make their too-long-delayed sojourn from offices to homes. Now, in 1982, I believe that home computers will literally 'take over' from these less sophisticated amusement toys.

"The next giant step forward spurring the growth of home computers is here," continued Spencer. "Home computers have become so popular that not only are computer-oriented retail outlets carrying them—many of the best known and most prestigeous department stores and audio/visual merchants are offering them too. Many new magazines and clubs for home

computer users continue to spring up regularly. And, as a direct result of this boom, more and more new services, such as data bases, educational programs, financial information, elctronic newspapers, and even 'shop-at-home via computer' are being offered to home computer users regionally and nationally.

"In serving this newly-created home computer public, Commodore has made sure that entire home computer systems are not only readily available, but available within financial reach. We are offering educational, recreational, and entertainment cassettes in six-packs for as little as $59.95. These include areas such as personal finance, a 'mini' word processor for use with the low-priced VIC 1515 graphic printer, a loan and mortgage calculation program, a home inventory program, and several others.

"Seven new cartridge games for wholesome low-cost family recreation at home are now available for only $29.95 each," added Spencer, "and many new games and educational programs are currently being developed.

"And, because computers—including home computers—are here to stay, it's important that future generations learn to program. Commodore has created a new "Introduction To Programming" package which allows VIC 20 users to learn right at home. It includes two cassettes and a home study text for only $24.95. And we have a single disk drive to add storage capacity to the VIC, and other low-priced memory expansion devices as well.

"The boom in home computers is here for sure, and it's getting bigger every day," said Spencer. "In fact, in 1982, Commodore alone will manufacture and bring to market as many computers as the entire microcomputer industry did in 1981. Surely that is not only dramatic growth, it's dynamic!

"The future of home computers is indeed bright," Spencer concluded, "and Commodore fully intends to be its guiding light!" ■

8K Memory Expander Introduced for VIC 20

The VIC-1110, a memory expander for the VIC 20, has been introduced.

The VIC-1110, which retails for $59.95, plugs directly into the memory expansion port of the VIC 20. The plug-in cartridge increases program capacity and adds 8,192 bytes of memory to the VIC 20 system.

In addition to the VIC-1110, a 3K memory expander (VIC-1210) is also available from Commodore at a cost of only $39.95. ■

Commodore to Offer Widest Range of Home Computers in the Industry with Addition of Two New Programmable Color Entries in Spring '82

Commodore Business Machines, Inc., will offer the widest range of home computers in the industry starting late Spring when it plans to introduce two new programmable/color low-priced home computers to compliment its line which currently is flagshipped by the VIC 20™.

According to Kit Spencer, Commodore's vice president-marketing, the two new home computers—the "COMMODORE ULTIMAX" and the "COMMODORE-64"—will play video games, compute, and even synthesize music through hook ups with stereo and hi-fi systems! They were demonstrated in prototype form for the first time at the Consumer Electronics Trade Show in Las Vegas.

"These two new exciting home computers will join the VIC 20 in offering the public capabilities unmatched for the price," said Spencer.

The COMMODORE ULTIMAX, which is planned to be introduced at a suggested retail price of $149.95, is a programmable color computer which will compete head-on at the retail level with less sophisticated video games such as the ATARI VCS and INTELLIVISION which do not offer computing capability. Easily connected to any color TV set, the COMMODORE ULTIMAX will feature a flat membrane keyboard, programming in BASIC, and use both cartridges and cassettes for games, programming, and music synthesis. The COMMODORE ULTIMAX will be compatible with joysticks, paddles, and light pens, use the VIC Datasette tape recorder for program storage, and serve as a sound generator for polyphonic tones and for music synthesis.

The COMMODORE-64, which will have a suggested retail price of $595.00, is an advanced home computer which will compete with ATARI 800 and the APPLE II+ among others. It is programmable with a CP/M compatible option, and has 64K of memory, well in excess of the competition in its immediate price range.

The COMMODORE-64 will have a full 66-key typewriter keyboard with upper and lower text capability, function keys, graphic characters, "smart" peripheral devices, a cartridge game slot, and game controllers. Its audio capabilities will include sound generation, polyphonic tones, music synthesization, and hi-fi output.

"This tremendous leap forward in home computer technology at competitive pricing is yet another advancement made possible by Commodore's unique in-house capability in designing and manufacturing microprocessor 'chips'," said Spencer. "We are the only microcomputer company that has this capability, and, in fact, chips of our design are currently being used today by many of our competitors worldwide.

"With home computers becoming more and more popular each day and truly becoming 'fixtures' in the Americana home, Commodore expects to manufacture and bring to market as many computers in 1982 as the entire computer industry did in 1981! We are that far ahead in technology, and the price/performance ratio offered by our diverse and comprehensive line of home computers is simply unsurpassed in the industry today.

"And, with the American public becoming more and more knowledgeable about home computers and about their many advantages over far less sophisticated and capable video games, the future for Commodore is indeed a bright one!" ■

New Teach-Yourself-Programming Series Introduced for VIC 20

The first in a series of teach-yourself-programming courses, Introduction to BASIC Programming, which provides a thorough introduction to BASIC programming on the VIC 20, has been introduced by Commodore.

Introduction to BASIC Programming comes complete with a self-study guide and two cassette tapes containing sample programs that run on the VIC 20. Even though the course relates to programming on the VIC 20, the concepts represented can be transferred to any other computer system, large or small.

The course is split into 15 units, each taking an evenings' work. Most units require reading, practical work with the VIC 20, and some programming. Experiments included within each unit reinforce the concepts presented. A self-test questionnaire at the end of each unit measures how well the unit was understood. ■

Commodore Restructures Marketing Organization to Gain Penetration of Home and Professional Markets

In a major plan to further expand its penetration of the booming home and professional microcomputer marketplaces, Commodore has restructured its marketing organization.

Kit Spencer, vice president-marketing of Commodore's Computers Systems Division, announced that two new divisions will be created and will be charged with marketing the company's four levels of home and personal computers.

The new Home Computer Sales Division will market Commodore's low-priced line of home computers, including the popular full-featured color VIC 20™, as well as other new products planned for 1982. Commodore's Professional Computer Sales Divisions will market the company's PET®, CBM™, and SuperPET computers.

Concurrent with the announcement of the restructuring, Spencer further made public that Bill Wade, who has been with Commodore since the mid-1970s when the company pioneered the microcomputer, has been named national sales manager for the Home Computer Sales Division.

"The marketplace for both home and professional microcomputers is highly volatile and has shown growth and growth potential that is simply astounding," said Spencer, "and we felt that segmenting our marketing organization was important for our continued penetration growth.

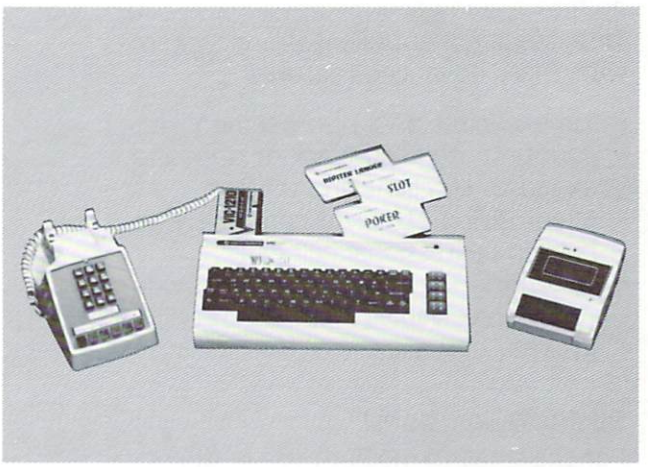
"There are four segments to the micro market," Spencer said, "home and hobbyist, educational, small business and professional, and the 'traditional' computer marketplace.

"Certainly the home and hobbyist market is showing giant potential, and we feel it's vital to bridge the gap between the game-playing video machines and home computers that also plays games, such as our VIC 20. This will become increasingly important as the public becomes more sophisticated and knowledgeable with regard to understanding and purchasing home computers. Part of our marketing plan will be to serve as not only an industry pace-setter, but as a reliable source of information for those seeking help in buying home computers.

"By restructuring our organization for future growth we will be able to gain wider distribution of the VIC 20 and our other home computers in this market. This will also allow us to concentrate our other efforts and resources on maintaining and increasing our market share in selling the PET, CBM, and SuperPET computers through the Professional Computer Sales Division, which we will continue to reach through a nationwide network of full servicing computer and business machine dealers who are served by our regional sales organizations.

"Commodore is in a very enviable position in our industry for several reasons," Spencer added. "This is due in part to the fact that we are the only micro manufacturer offering four levels of hardware at four distinct price points—and we will be offering even more innovative hardware in the very near future. Another key factor helping us gain this competitive edge is that again we are the only full-integrated micro manufacturer, and that gives us a big advantage, not only in designing and manufacturing micros, but also in offering a price/performance ratio unsurpassed in our industry." ■

Commodore Introduces Lowest Priced Modem in the Computer Industry



The new "VICMODEM," which retails for $109.95, is an easy-to-use plug-in cartridge that connects directly to the user port of Commodore's VIC 20™ home computer, and may be used with any modular style telephone.

The VICMODEM, which is planned for retail sale in the Spring of 1982, allows users to communicate and exchange data with other computer owners over the telephone. This latest VIC peripheral also allows users to inexpensively access telecomputing networks such as Source™ or CompuServe™, which provide services such as stock quotes and company reports, newswire stories, research data, sports scores, airline reservations, shopping services and more.

The VICMODEM is a direct connect, 300 baud modem with originate/answer and half/full duplex capabilities. The combined cost of the VIC 20 and a VICMODEM is less than $410.00, while some individual modems retail for over $400.00. ■

The Information Solution To Your Office Productivity Bottleneck!

Professional Solutions!

For the attorney, consultant, engineer, architect, and other professionals . . . a Time Management/Client Billing System with these advanced features:

- Timekeeping and Disbursement Accounting - to keep track of time and expenses by client and matter.
- Client Billing - to print computerized bills on professional billheads.
- Accounts Receivable - to provide Aged Receivables reports for improved "cash flow" control.

Business Solutions!

For the small to medium size business... a comprehensive Management Accounting System with these modules:

- Sales Order Entry
- Inventory Management
- Accounts Receivable
- Accounts Payable
- Payroll
- General Ledger

Now . . . turn your Commodore computer into a powerful business system with Info-2001™ Software Applications.

Info-2001™ Software Applications are designed by expert consultants to meet the needs of the professional practitioner and businessman. All systems have been thoroughly field-tested at many business sites to insure complete functionality.

The "parameter-driven" design allows you to customize the system to conform to the specific practices of your office. You get the best of both worlds—a "field-proven" packaged system (at off-the-shelf prices) with the "flexibility" to meet your specific needs.

All software includes an instructionally designed Operator's Manual, functional input forms, and training by your authorized dealer using a computer-simulation model.

For a demonstration of these "Information Solutions" see your local authorized Commodore dealer today!

InfoDesigns

INFORMATION SOLUTIONS USING MICROCOMPUTER TECHNOLOGY

Info-Designs, Inc. • P.O. Box 5340 • W. Bloomfield, Michigan • (313) 582-9090

Market Research Firm Uses 8032s to Streamline Operation

We've all had the experience: the phone rings and it's someone doing a market study, or we are approached in a shopping mall to answer "just a few questions." Most of the time we respond, but almost never do we have any idea about what's happened to our comments and our viewpoints.

Since being founded over four years ago, Market Analytics of Narberth, Pennsylvania, has run the gamut of these and other interviewing techniques in a successful effort to establish the company as an effective market research organization.

Stan Hunter, president of Market Analytics, recalled the obstacles his young company faced in processing vital research information.

"When we started in business," he said, "we realized that one of the major problems facing our organization was how to take a survey research project consisting of 'little marks on a piece of paper,' and then convert that information into meaningful data analysis."

Hunter's search for a remedy to this problem occurred almost by accident. Recognizing his need to generate research documents quickly and accurately, he first began looking at word processors. "We quickly came to the conclusion that to get a decent word processor, including the necessary peripherals, we would have to spend almost $20,000," he recalled. And too, Hunter realized that although these word processing machines used 16-32K of memory, they often performed no other functions.

Employing an independent consultant to pursue alternatives to a word processor, Hunter was advised to purchase a Commodore CBM 8032. In the beginning, the office staff operated the computer primarily as a word processor, using WordPro 4 Plus from Professional Software, Inc. of Needham, Mass. "It has worked out very very well for us," said Hunter. "We often give our clients the opportunity to review drafts of our research reports, because we need to know if we've touched all bases. Before our word processing system, it was brutal to have to go back and revise a 250-page report," he said. "Now, it's so easy—why it almost scares me."

Having satisfied his word processing needs, Hunter decided it was time to utilize the capabilities of the 8032 to its fullest, and began considering the alternatives in choosing a data analysis system. Realizing that he would require highly sophisticated techniques and rapid turnaround, Hunter opted to go with a computer service company that offered timesharing alternatives. "We did a very extensive search of interactive computer systems and decided that Comshare was by far the best operation for our specific needs," said Hunter.

Comshare, headquartered in Ann Arbor, Michigan, is a computer service company offering data analysis with techniques such as basic statistics, regression analysis, factor analysis, cluster analysis, crosstabula-

→

tions, survey analysis, and hypothesis testing. Concurrent with his decision to use the Comshare network, Hunter also purchased a second CBM 8032.

There are several ways to link up with the Comshare network. These include keypunched cards, paper tape, or interacting through a terminal. Because his firm already had two 8032s, Hunter chose the last method, knowing he could use his computer as an interactive terminal.

Hunter admitted that it would be faster if he was working directly on a CRT terminal. However, he pointed to the cost advantage, noting that his two 8032s cost less than $2,000 each. "One CRT is close to $30,000," said Hunter, "so it is obviously far less expensive to go with the CBM." "Plus," he candidly admitted, "we have a word processor and can even play space games!"

Although Comshare could both create and process the market research data for Market Analytics, Hunter had to confront the very expensive problem of constantly using the phone line to interact with the Comshare computer.

Therefore, Hunter decided to employ the services of John Odgers, of Applied Information Design Systems of Maple Shade, New Jersey, to create a cost-effective program that would translate research data into meaningful statistical format before it was transmitted to Comshare. Equally important to Hunter was ease of use. "I wanted the system designed to the point where the information could be handled by clerical people," said Hunter, "and John has done just that."

"We wanted to control everything here—create the data, define the survey, enter the data, and verify the data. That way, when we go interactive with Comshare, we're going 'gangbusters,' because they are getting clean data."

Happily posing with one of three CBM 8032s used for word processing and market research are, clockwise from left: Market Analytics president Stan Hunter, John Odgers of Applied Information Design Systems, Comshare representative Mitch Rubin, and Market Analytics' data processing manager Beverly McIntire.

A Survey

Basically, Market Analytics' system allows the user to define a survey, enter the questions, prepare the data for Comshare, and analyze the data. When the surveys are defined, each question is assigned a code number. A project director will then enter a number for each answer received and the computer will translate these codes into readable information.

The advantage of working with codes, according to Hunter, is that codes are all assigned numerical values. And Hunter feels that the CBM 8032 offers a distinct advantage with its 10-key numeric pad. "The numeric pad allows the operator to use a 'touch-typing' system, meaning you never have to take your eyes off the questionnaire as you are asking questions and entering answers," said Hunter.

Because the success of any survey depends upon the accuracy of the information, Hunter enters all data a second time for verification. In the past, using the old method of key verification, an operator would punch all information onto cards. Then another person, using a key verifying machine, would punch the same exact data to ensure its accuracy.

According to Beverly McIntire, who is responsible for the firm's data processing, the new system has happily put an end to this duplication of effort. "Now when a mistake occurs," McIntire explained, "the system flags the error and gives us choices for correcting the problem. With the old manual system, the choices were never made apparent to the operator."

"We still enter the data internally and control the information internally," added Hunter. "Nothing leaves the office. We are using the computer for everything we could possibly perform internally. And we are using Comshare only for processing information. We could also verify and update our data using Comshare but that gets expensive. Having total control over the information saves us tremendous transmission time."

Market Analytics is also eagerly anticipating the arrival of a third

CBM 8032 very shortly. Although Hunter could easily use less computers for his data analysis requirements, he generally employs one 8032 strictly for word processing. Still, to expedite results, Market Analytics' system is designed so that three operators can be working on one survey simultaneously. Hunter explained that information from completed questionnaires is entered on two computers. After the information has been entered, the data is merged onto one disk for transmission to Comshare on the third CBM.

Before transferring information, Hunter loads PETTERM* into the computer, which converts the CBM into a terminal, and allows Hunter to access the Comshare network.

Market Analytics transfers their data to Comshare using their 300 baud modem. If necessary, Hunter plans to upgrade his modem to 1200 baud. Depending on the need for immediate feedback, Comshare can process the information and return it interactively, or the data can be batch processed overnight, which is less expensive.

After the data has been analyzed by Comshare, it is returned by the telephone network to Market Analytics, where it is downloaded to disk. From that point, the information is retrievable from disk, thus reducing the cost of interactive transmission time.

All data is dumped on cassette after it has been processed, and is stored in safety deposit boxes. "Even the money we save storing these cassettes is obvious," said Hunter. "You can fit a lot more cassettes in a safety deposit box, rather than storing large 200 page reports."

According to Hunter, "The time we have saved using this system has been astronomical." In their first attempt at using the new system, Market Analytics completed a 10-day job in just three days. Hunter estimated that with an average survey, his firm will now save "a minimum of 30 percent completion time while generating far more accurate results."

So, the next time you're the subject of a market study, your responses may very well be tabulated, adjusted, analyzed, and processed by a Commodore microcomputer. ■

—Paul Fleming

*PETTERM is a registered trademark of FSS Company, Austin, Texas

The best game in typesetting—use a personal computer for input

By Naomi Blackburn
Partner, Blackburn Associates
La Crescenta, California

Personal computers are famous for the games they play—Space Invaders, Star Trek, Chess, and so on. And, while preparing material for typesetting is no game, these versatile, low-cost computers, along with suitable programming and accessories, can make the work seem almost like play.

My husband and I toyed with trying to develop a front-end system for our Mergenthaler V-I-P using a personal computer ever since they appeared on the market. But the complexities of programming, interfacing, and integrating a computer system into the operation that provides our livelihood on a learn-as-you-go basis kept us from making the attempt.

Recently, however, we discovered that Type Share, a Downey, Calif., company had done the development for us. And since that time, we have been enjoying a level of speed and ease of production that we could have attained only with a high-priced, packaged front-end system in the past.

Like many in typesetting, we grew up with the cold-type industry, starting part time with a Varityper and advancing to a punched-tape system using Justowriters. Initially, we rented time on a phototypesetter (to run our tapes) and eventually, after struggling with used equipment, we bought a new V-I-P with programming to accept Justowriter tapes as input. This provided us with a reliable, high-quality typesetter at the same time that it allowed us to continue using our Justowriters to front end the operation.

Thus, we never could see our way clear to spend the $10,000 plus that the typesetter manufacturers charge for their computerized front ends, especially since the computers in these units are dedicated—there is no way to use them for other applications, such as bookkeeping, mailing, and so on.

But as time went by, an increasing volume made our punched tape system more and more unwieldy, the Justowriters began to break down more often, and getting parts became a problem. So when personal computers appeared, we immediately looked into the possibility of adapting them to our requirements. And, while we were not able to put a system together ourselves, the Type Share system works almost as though we had.

The system is built around the Commodore CBM 8000 series computer and includes a dual drive CBM diskette unit, a tape punch, software, and an optional CBM line printer. It handles keyboarding, editing, making corrections, and coding for typesetting. (At present, the system works with V-I-P's and Photon's and it is being adapted to other typesetters as the need arises.) And it costs less than half of what many typesetter manufacturers charge for their dedicated front ends.

Beyond this, the system is much more versatile than many packaged front-end systems because of the fact that the CBM is a general-purpose computer. For example, editing functions can be applied to an entire file, not just the lines on the screen. Files can be merged or divided and recorded on disk as new files. Global search and substitute functions aid in reformatting entire files with a few keystrokes. And, of course, repetitive jobs can be retained on disk so updates amount only to simple insertions and changes.

Further, because the system outputs punched tape, no alteration or interface is required for our typesetter. Thus, the Type Share front end is ready to go to work when it is delivered and, as far as the typesetter is concerned, nothing has changed. However, the typesetter is doing more work each day and we are doing less as the Type Share system speeds input. There are several reasons for this improvement:

- The comfortable, quiet, electronic keyboard of the CBM makes the initial keyboarding for jobs fast and easy, while software features like automatic carriage return and automatic listing of copy for review and proofing speed the work.

- Changes and corrections are fast and easy, too, because simple commands allow for changing lines, inserting copy, moving blocks of copy, and making substitutions within lines.

- Proofing is aided by simple commands that list the next lines in a file, list the same lines automatically after changes, and list previous lines (essentially a scroll down function). Further, with the inexpensive line printer, the system can produce a hard copy on cheap, line-printer paper (instead of having the typesetter set a proof on expensive phototypesetting paper) for proofreading and review of typesetting codes.

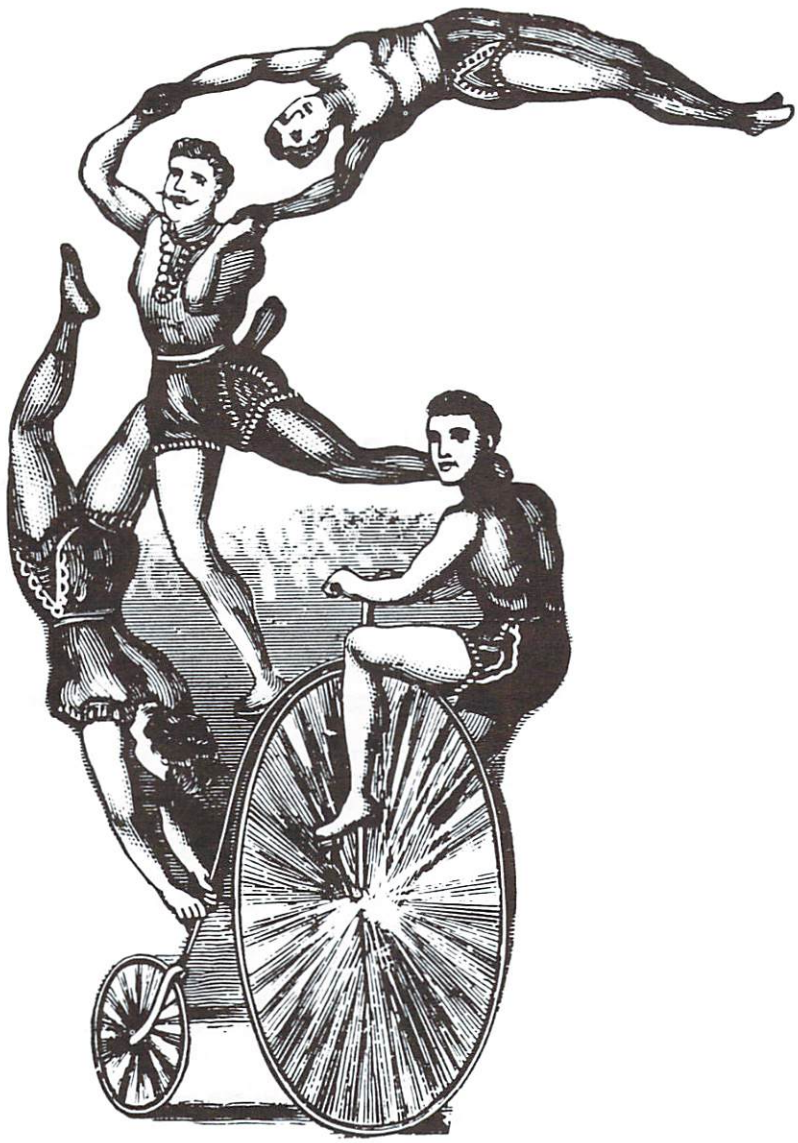
• Reading and writing disk files is handled by the software. In fact, virtually all computer operations are handled by the program so no knowledge of computers or programming is required to operate a Type Share system.

• When a file is ready for typesetting, a command from the keyboard activates the punch and the entire file is punched without further operator intervention.

But the big plus for the system is the fact that the general-purpose computer is available for other work when it is not needed for typesetting. A variety of bookkeeping, accounting, and management information system software is available for the computer so, simply by loading another program, the system can help us with other aspects of our business. And, by the way, it also plays great games. But the best game is the way it handles input for typesetting—now that we have it, we cannot remember how we got along without it. ■

Reprinted with permission from TYPEWORLD Magazine, December 11, 1981

Perfectly Balanced



educational software

from MICRO-ED for PET® and VIC®

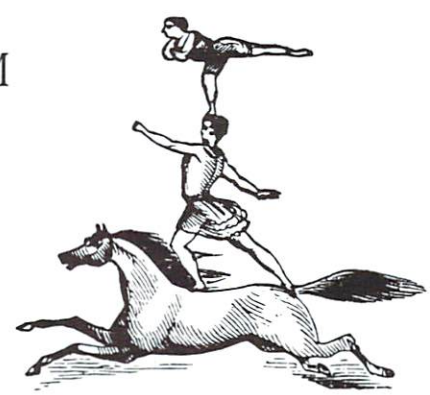
Send for our free catalog*

*please specify PET or VIC

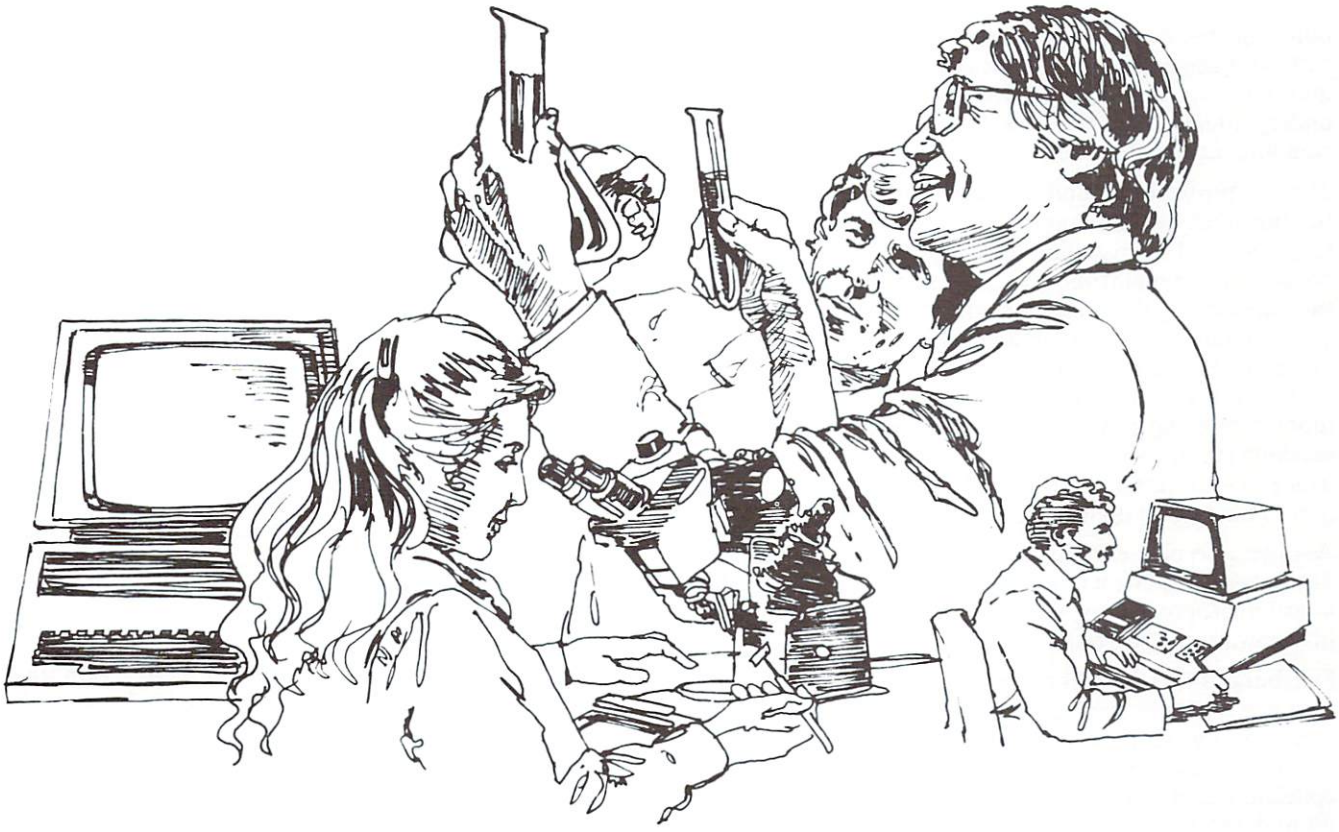
MICRO-ED, Inc. • P.O. Box 24156—COM

Minneapolis, MN 55424

or telephone us at (612) 926-2292



VIC and PET are the registered trademarks for Commodore Business Machines.



PETs Provide Solution in Chemical Engineering Lab

Computers are beginning to be used widely in education, but perhaps nowhere are they being used more effectively or economically than in the undergraduate chemical engineering laboratory courses at the University of Rochester.

In the University of Rochester's undergraduate chemical engineering laboratories, more than two dozen Commodore PET microcomputers are being used routinely to collect and process data in the course of performing standard undergraduate experiments. They are also being used to perform calculations on this data, using computer programs written by the students.

"This process has altered the fundamental nature of our laboratory," said Richard H. Heist, associate professor of chemical engineering. "Using these computers enables our students to concentrate more on the chemical engineering principles behind an experiment rather than the actual mechanics involved in carrying it out. At the same time the students learn important problem-solving concepts and become familiar with techniques and equipment that are increasingly common in industry."

It was Heist, his colleague Howard Saltsburg, professor of chemical engineering, and Thor Olsen, supervisor of the undergraduate laboratories, who equipped the chemical engineering laboratories with microcomputers. Up to three years ago none of them had ever used these devices, but they were aware of the increasing importance of com- →

puters in industry and education. Accordingly, Heist and Saltsburg began looking for a way to integrate computers into the academic environment of the students at the undergraduate level. "We wanted students to regard the computer as a tool," said Heist.

The two professors found that the most common method for introducing the computer into the teaching laboratory consisted of installing terminals that were remotely connected to a mainframe or minicomputer. This system, they decided, had a number of disadvantages: the physical separation of terminal and computer and the "detached" nature of the system; the lack of obvious local control by the students; the availability of much more processing power than is normally needed in the teaching laboratory; the cost; and finally the total dependence of each terminal on a single unit (if the main computer failed, everything went down).

Another alternative was to "build" a computer from fundamental components, but this method, using single-board microprocessors, requires extensive expertise both in hardware and software.

Saltsburg and Heist recognized that the completely self-contained, simple, and relatively inexpensive PET microcomputers were ideally suited to the requirements of chemical engineering experiments for extensive, routine collection and processing of data. "Students are less intimidated by a device for which the power switch is within easy reach. They can quickly develop confidence and capability using such a device," says Saltsburg.

Although less powerful than larger minicomputers, the PET is intrinsically more convenient to use for data acquisition and process control. (Process control is an important industrial application in which information based on collected data is used to control temperature, pressure, flow rates, and other variables in an industrial process.)

In addition to offering economy and simplicity, these personal computers have other important advantages in both hardware and software, according to Heist and Saltsburg. Each can control many input and output devices, including a variety of laboratory instruments such as temperature and optical sensors, digital multimeters, and power supplies. "The immediate availability of a high-level language makes the transition to computer usage much simpler," said Saltsburg.

According to faculty members, the speed and efficiency of PET-controlled data collection allows students to spend most of their time on analysis and interpretation. According to John Friedly, Chairman of the Department of Chemical Engineering, this more efficient use of laboratory time has enabled the chemical engineering department to meet the needs of an undergraduate enrollment that has more than doubled in the last five years. At present there are more than two dozen microcomputers in use in the undergraduate laboratory; they serve more than 210 students.

An important feature of Rochester's microcomputer-based laboratory program in chemical engineering is its emphasis on the fundamental principles of computing. Through the extensive participation of F. W. Arcuri, University Computing Fellow in the River Campus Computing Center, students are exposed to programming concepts and notations that not only are useful in the laboratory but also, Heist said, "are valuable in teaching the thought processes and problem-solving strategies necessary for engineers and scientists to perform their jobs effectively."

After exposure to BASIC, the students are currently taught FORTH and PASCAL. These programming languages, Heist said, demand that students apply the very techniques that are important to problem solving: a careful definition of a complex problem and its breakdown into smaller, simpler subunits.

According to Friedly, several universities have already asked the University of Rochester group for advice on setting up computer-aided undergraduate laboratory courses of their own. In responding to these requests Heist and Saltsburg emphasized that the philosophy behind Rochester's successful program has been and continues to be "to use the computer only when there is a significant gain in engineering content and to keep everything simple, cheap, and reliable." ■

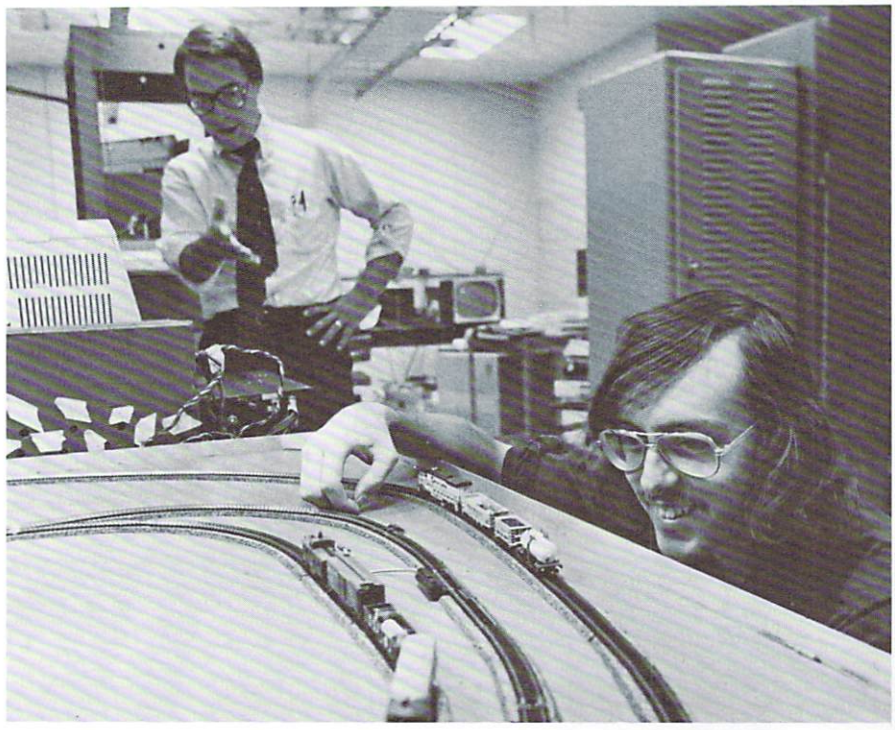
Students meet PETs in engineering laboratory

In a ground floor laboratory in Gavett Hall, two model trains approach each other on a collision course. Suddenly and automatically, they change speed or switch tracks to avoid crashing into each other.

Hardly a game, the trains are among the many tools used to teach chemical engineering students how to use computers to control experiments. In this case, a PET microcomputer was programmed to check the trains' locations, using phototransistors embedded in the track, and to control the engines' speed and direction.

The University's chemical engineering graduates probably won't find model trains on the job, but "the principles our students learn have broad application in industry, where the use of microcomputers has grown enormously in the past five to seven years," said Richard H. Heist, associate professor of chemical engineering. "With increasing miniaturization of computer circuitry, equipment prices have dropped while capabilities have gone up, and it has become cost-effective for companies to use microcomputers in routine, everyday operations," Heist explained.

Increasingly, industrial firms are using microcomputers not only to acquire data, but to feed back information based on the data to control the industrial process itself, Heist said. "Computers can be programmed to make decisions concerning a process and then to operate switches to control temperature, pressure, flow rates, and a host of other variables," he added. "'Process control' by computers frees the engineer from routine, time-consuming procedures and increases overall productivity." Integrating the computer into the laboratory environment also "enables our students to spend more time learning chemical engineering and less time doing repetitive manual measurements and



operations," he said. "And a computer's attention doesn't wander during a long experiment, like that of a tired student."

In the last two years, the University's chemical engineering department has become one of the "very few" departments nationwide to equip their undergraduate laboratories extensively with microcomputers, according to Heist. As sophomores, all chemical engineering majors at Rochester must take a laboratory course that covers aspects of computer programming and the relationship of programming to the concepts needed for analytical thinking and problem-solving, as well as techniques for connecting computers to laboratory equipment. Students use microcomputers, rather than larger, more complex computers "because the smaller, self-contained machines are less intimidating and more likely to build student's confidence in their use," Heist said. "Also, they are relatively simple to connect to laboratory equipment."

Chemical engineering students learn a modern computer language, FORTH, which is particularly suited to process control. Unlike statement-oriented computer languages like BASIC and FORTRAN, FORTH provides a mathematical style of programming that allows students to divide problems into simpler units, according to University Computing Fellow William Arcuri, who teaches the sophomore laboratory course with Heist and Prof. Howard Saltsburg. Interested students can pursue advanced study in FORTH, making them highly attractive to employers. "One senior last year became especially knowledgeable and landed an excellent job with IBM, largely because he had this expertise," Heist said.

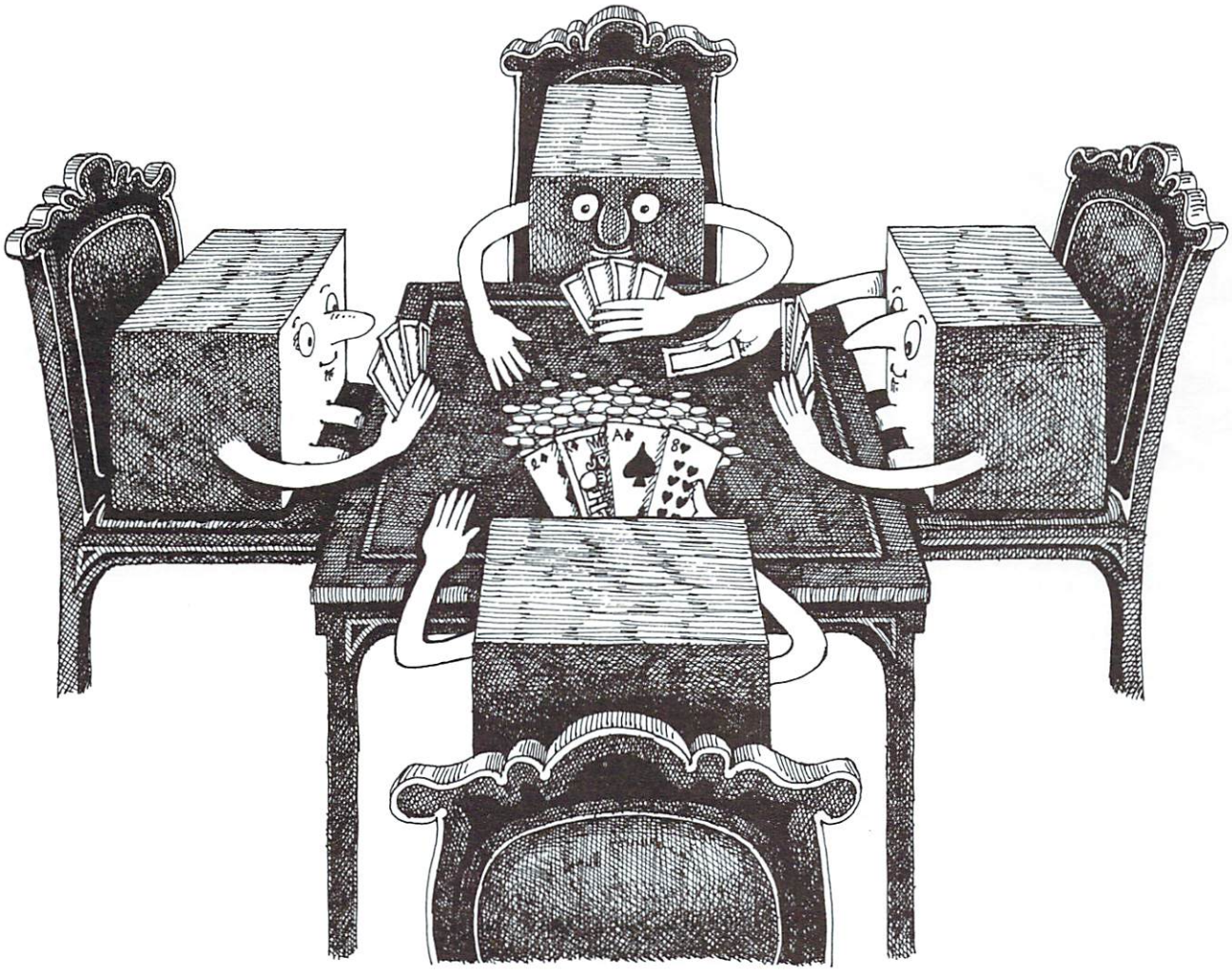
With about three dozen microcomputers available in the department, Chemical Engineering has made the most extensive teaching use of microcomputers of any River Campus department, according to Heist.

→

Although Chemical Engineering probably has done the most teaching in process control, faculty in other departments have had students use, or are planning student use of, microcomputers for other academic purposes. These departments include Chemistry, Biology, Mathematics, Electrical Engineering, Mechanical Engineering (to solve actual design problems as well as to collect and analyze data), Computer Science, and English (to teach composition skills). The Graduate School of Education and Human Development also has established a microcomputer laboratory and has begun to offer courses covering computer applications in elementary and secondary school curricula.

These expanded offerings reflect a national trend in microcomputer use at home, in high schools, and even in some elementary school classrooms, according to Arcuri. "An increasing number of our students have had computer experience before they arrive at the University," he said, "and we're certain to see more interest in this area among students in future classes."

"These new teaching uses of microcomputers fit the overall pattern of growing—and varied—opportunities on campus for Rochester students to work with computers," added Sidney Shapiro, professor and chairman of the Department of Electrical Engineering. "In addition to microcomputers, students also use minicomputers, microprocessors, and large machines such as the IBM 30-32, accessed by terminals," he said. ■



Smart Boxes

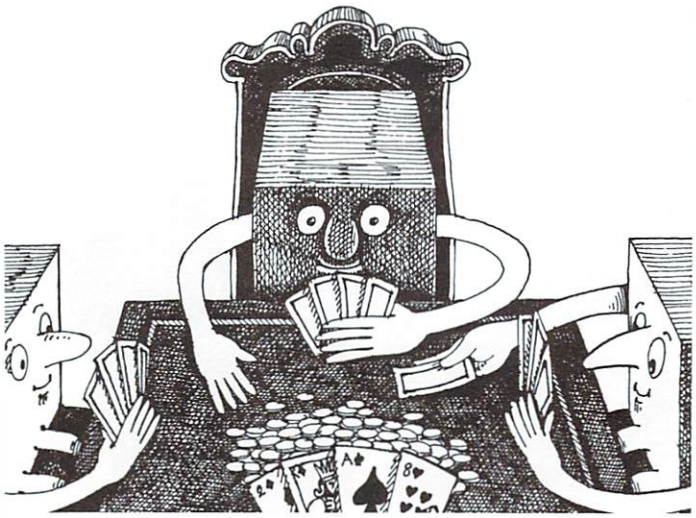
by
Prof. D. G. Barker
Dept. of Educational Psychology
Texas A&M University

In *The I Hate Mathematics Book* (Boston: Little and Brown, 1975), Marilyn Burns and Martha Hairston describe a simple "magic trick" based on the binary number system. The trick requires the preparation of five cards, each containing 16 numbers. The numbers are between 1 and 32, inclusive. (Obviously, some numbers appear on more than one card.) The magician asks someone to select a number between 1 and 32 and then to point to each of the cards that contains the number selected. The magician then tells the audience which number had been selected.

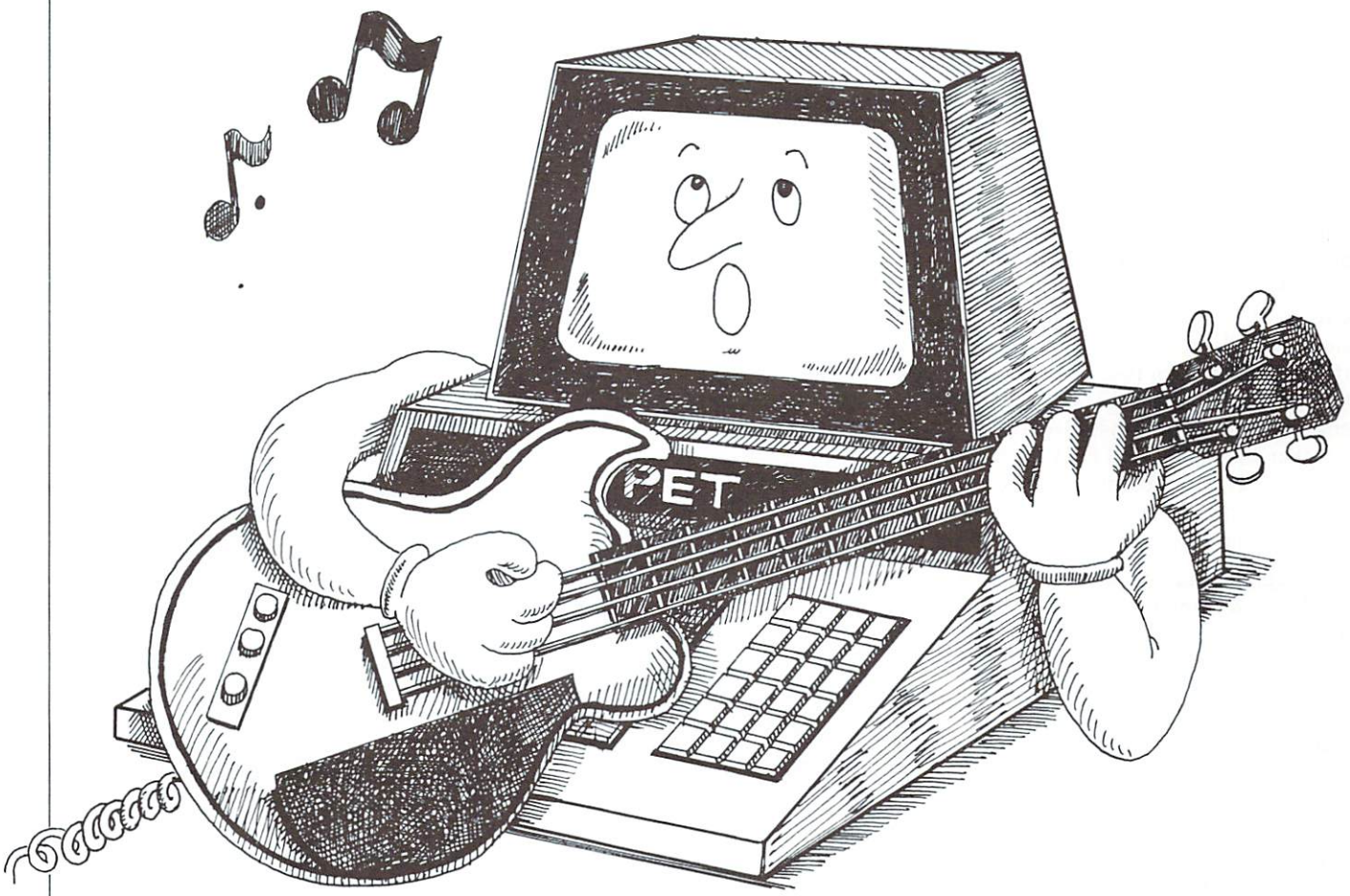
The performer does this trick by summing the first number on each card the victim confirms as containing the number selected. All decimal numbers from 1 to 32 can be represented in binary by five or fewer binary digits (0 or 1). The first card contains a 1 as its first number and all the decimal numbers that contain a 1 rather than a 0 in their first (right-most) digit when transformed to binary. The second card contains a 2 as its first number and all the decimal numbers that contain a 1 in their second digit when expressed in binary. Similarly, the third through fifth cards begin with the second through the fourth powers of two (4,8,16) and all the decimal numbers that contain a 1 in the third, fourth, and fifth positions, respectively, when expressed in binary.

The program "SMART BOXES" is an implementation of this trick for the Commodore PET computer. It instructs the user to select a number between 1 and 32, prints the cards or "boxes," asks the user if his or her number is in each box in turn, and then announces the number previously selected by the user. This little program can stimulate children's interest in learning the binary number system, on which all digital computing is based. ➔

EDUCATION



```
100 PRINT TAB(15);"■■SMART BOXES"
110 PRINT"THINK OF A WHOLE NUMBER"
120 PRINT"  BETWEEN 1 AND 31."
130 PRINT"  WHEN YOU HAVE IT, PRESS ANY KEY."
140 GET R$:IF R$="" THEN 140
150 PRINT"■"
160 GOSUB 1000
170 N=0
180 FOR I=1 TO 5
190 PRINT"IS YOUR NUMBER IN BOX #";I;"?";
200 R$=""
210 GET R$:IF R$="" THEN 210
220 IF R$="Y" THEN PRINT "YES"
230 IF R$<>"Y" THEN 300
240 IF I=1 THEN N=N+1
250 IF I=2 THEN N=N+2
260 IF I=3 THEN N=N+4
270 IF I=4 THEN N=N+8
280 IF I=5 THEN N=N+16
290 GOTO 310
300 PRINT"NO"
310 NEXT I
320 PRINT
330 PRINT"YOUR NUMBER IS ";N;"■."
340 PRINT"PLAY AGAIN?";
350 R$=""
360 GET R$:IF R$="" THEN 360
370 IF R$ ="Y" THEN PRINT "YES.":GOTO 100
380 PRINT "NO"
390 END
400 REM--BASED ON "I HATE MATHEMATICS"
    BY BURNS AND HAIRSON, P. 70.
1000 REM--SUB TO PRINT BOXES
1010 PRINT "  BOX #1          BOX #2"
1020 PRINT " 1  3  5  7       2  3  6  7"
1030 PRINT " 9 11 13 15      10 11 14 15"
1040 PRINT "17 19 21 23      18 19 22 23"
1050 PRINT "25 27 29 31      26 27 30 31"
1060 PRINT
1070 PRINT"    BOX #3          BOX #4"
1080 PRINT" 4  5  6  7       8  9 10 11"
1090 PRINT"12 13 14 15      12 13 14 15"
1100 PRINT"20 21 22 23      24 25 26 27"
1110 PRINT"28 29 30 31      28 29 30 31"
1120 PRINT
1130 PRINT"            BOX #5"
1140 PRINT"         16 17 18 19"
1150 PRINT"         20 21 22 23"
1160 PRINT"         24 25 26 27"
1170 PRINT"         28 29 30 31"
1180 PRINT
1190 RETURN
READY.
```



PET Musician Plays BASIC Rock N' Roll

A recent electronic music festival featured several bizarre bands, but it wasn't a guitar or a keyboard that stole the show—it was a computer!

The event was the second annual Festival of Performing Philadelphia Electronic Musicians at Temple University, and the computer was a Commodore PET used by a band called (no, not the Commodores) the "Rubberheads."

George Kuetemeyer "plays" the computer and other assorted electronic instruments for the Rubberheads, whose other members include singer and songwriter Scott Lomba on guitar, Mike Gunning on guitar, and Jeff Lomba (Scott's seven-year-old son) on keyboards. Kuetemeyer usually uses the PET for a base and percussion effect and sometimes even hooks up a voice synthesizer for back-up vocals.

Kuetemeyer says that the voice effect is a real crowd pleaser. He programs the computer so that keys can be assigned a specific word. He can then push a key at the appropriate time in a performance for the desired effect. For example, he can assign the B key on the PET keyboard to say "baby, baby." Then, when the singer says "baby," Kuetemeyer pushes the B key and the computer backs up the singer for an amazing effect. A "star" though it may be in the Rubberhead's act, Kuetemeyer admits that the PET "is not quite ready for lead vocals." (See program following this article.)

When he is not thrilling crowds with his unique music, Kuetemeyer coincidentally works in the service department at a Commodore dealer, A. B. Computers in the Philadelphia suburb of Colmar. He started playing electronic music with a home-built analog sequencer. "it was fun but at that time still impractical for use in performances," said Kuetemeyer. "Now the effects I create with the PET would be impossible with the sequencer."

Using the PET allows Kuetemeyer to store music on tape and play up to a 40-note sequence. Kuetemeyer's 32K PET is hooked to a variety of electronic gadgets including an anolog and percussion synthesizer. The IEEE port is used to initiate sound events, and the CB2 line of the user port provides the anolog synthesizer with pitch information. A special sequencer program turns the PET

→

into both a drummer and a bass player for the group.

Kuetemeyer chose the PET over other computers for a variety of reasons. "The PET is all-in-one including a CRT so I don't have to drag a television around," he said. Other features that he enjoys include the built-in tone generator and CB2 sound which enable him to program in BASIC, yet another benefit of the PET because many other microcomputers use far more complicated and time-consuming machine language.

Although he has been pleased with the PET, Kuetemeyer is still experimenting with different ways to get a new or better sound, even by trying to use a Commodore VIC 20 in its place. According to this innovative musician, "the VIC and a small five-inch television can be packed into a brief case, making it even more portable than the PET." Kuetemeyer noted that another advantage of the VIC is on-board sound generation with four "built in" voices. Three of these "voices" are for tones while the fourth is a "white noise" generator that helps create the sound effects for the VIC's video games.

Computers have entered virtually all areas of every day life, and music may well be the next field where they make their mark despite the fact that some people resent computers being used in the Arts. They believe that it replaces man's creativity, and that taped or pre-programmed music takes something away from a live performance. Artists like Kuetemeyer, however, would argue this point, because although the music can be taped, it is manipulated at a live performance for a sound that is always unique and fresh. The computer can actually add to creativity by allowing the storage of an endless selection of variations of music on tape or disk.

Electronic musician George Kuetemeyer poses with the "Rubberhead's" drummer and bass player—a Commodore PET!

Photo courtesy Devine Studios

In offices, homes, and schools, computers have and are continuing to revolutionize the way we work, live, learn. And now they're even making music! Commodore's place as a leader in this revolution was born out at the festival when a young boy in the first row pointed at the computer being used by the Rubberheads on-stage and said aloud in amazement, "Hey mom, we have a PET just like that in school!"

— *John O'Brien*

VOXBOX

VOXBOX is a program designed to be used with a Commodore 4010 voice synthesizer and most Commodore PETs. For more information, contact George Kuetemeyer at A. B. Computers, (215) 822-7727.

```
READY.

 1000 REM *
 1001 REM *** VOXBOX 81 ***
 1002 REM *
 1003 REM * G. KUETEMEYER 12/12/81
 1004 REM *
 1010 PRINT"■S";:REM CLEAR SCREEN
 1011 PRINT"■■■■■■■■■■VOXBOX INSTRUCTIONS
 1012 PRINT"
 1013 PRINT"VOXBOX ALLOWS 4010 VOICE SYNTHESIZER
 1014 PRINT"OWNERS TO RECORD, PLAY AND SAVE UP TO
 1015 PRINT"10 SETS OF WORDS OR PHRASES.
 1016 PRINT"■PRESS THE ■W■ KEY TO SELECT A WORD SET.
 1017 PRINT"YOU MAY USE ANY DIGIT FROM '0' TO'9'
 1018 PRINT"■■■■■■■■■■■■■PRESS ANY KEY TO CONTINUE":GET K$:IF K$=""GOTO 1018
 1019 PRINT"■TO START RECORDING WORDS, PRESS
 1020 PRINT"THE ■R■ KEY. PRESS ANY KEY FROM 'A' TO
 1021 PRINT"'Z'. THIS WILL ASSIGN YOUR WORD TO THAT
 1022 PRINT"KEY WHEN YOU ENTER THE PLAY MODE.
 1023 PRINT"YOU WILL BE PROMPTED TO ENTER A WORD OR
 1024 PRINT"PHRASE OF UP TO 30 CHARACTERS. CONSULT
 1025 PRINT"YOUR INSTRUCTION MANUAL TO DETERMINE
 1026 PRINT"WHICH CHARACTER REFERS TO WHICH PHONEME.
 1027 PRINT"TO SPECIFY AN INFLECTION CHANGE,
 1028 PRINT"KEY IN A ■π■ FOLLOWED BY A '0','1','2'
 1029 PRINT"OR '3'. YOU MAY EXIT THIS MODE BY
 1030 PRINT"PRESSING THE 'RETURN' KEY.
 1031 PRINT"
 1032 PRINT"IN ORDER TO PLAY YOUR WORDS, PRESS THE
 1033 PRINT"■P■ KEY. WHEN YOU PRESS ANY KEY FROM 'A'
 1034 PRINT"TO 'Z',THE WORD THAT YOU ASSIGNED TO
 1035 PRINT"THAT KEY WILL BE PLAYED ON YOUR 4010.
 1036 PRINT"
 1037 PRINT"YOU CAN SAVE YOUR WORDS ON CASSETTE BY
 1038 PRINT"PRESSING THE ■S■ KEY. TO LOAD THEM BACK
 1039 PRINT"PRESS THE ■L■ KEY.
 1040 PRINT"
 1041 PRINT"PRESS ANY KEY TO BEGIN
 1042 GET K$:IF K$="" THEN GOTO 1042
 1050 REM *
 1051 REM *** INITIALIZATION ***
 1052 REM *
 1060 PRINT"■■■■■■■■■■■■■■■INITIALIZING STRING ARRAY ";:REM CLEAR SCREEN
 1070 DIM VX$(10,27):REM VOCAB ARRAY
 1080 FORI=0TO9:FORJ=1TO26:VX$(I,J)="A":NEXTJ,I:REM DEFAULT WORD ARRAY
 1090 VN=0:REM WORDSET DEFAULT
 1100 GOSUB 1920:GOSUB 2160:REM INITIALIZE 4010
 1110 GOSUB 1130:REM GOTO KYBD MONITOR
 1120 END
 1130 REM *
 1140 REM *** KYBD MONITOR ***
 1150 REM *
 1160 GOSUB 1680:REM PRINT WORD SET
 1170 PRINT"■■■■■■■■■■■■■■■■■■■■      ■P■LAY ■R■ECORD ■L■OAD ■S■AVE ■W■ORDSET#";
 1180 GET K$:IF K$="" GOTO 1180
 1190 PRINT"■■■■■■■■■■■■■■■■■■■■                                        ";
 1200 IF K$="R"THEN GOSUB 1260:REM RECORD
```

→

```
1210 IF K$="P"THEN GOSUB 1390:REM PLAY
1220 IF K$="L"THEN GOSUB 1520:REM LOAD WORDS
1230 IF K$="S"THEN GOSUB 1600:REM SAVE WORDS
1240 IF K$="W"THEN GOSUB 1760:REM SELECT WORD SET
1250 GOTO 1130
1260 REM *
1270 REM *** RECORD MODE ***
1280 REM *
1290 PRINT"SQQQQQQQQQQQQQQQQRECORD MODE  RRETURN TO EXIT          "
1300 PRINT"SQQQQQQQQQQQQQQQQQQQKEY(A-Z)";:GETKK$:IFKK$=""GOTO1300
1310 K=ASC(KK$):IFK=13 THEN RETURN
1320 K=K-64:IF K<0 OR K>91 GOTO 1300
1330 PRINT"SQQQQQQQQQQQQQQQQQQQQQ]]]]]]]";KK$;
1340 VS$=VX$(VN,K)
1350 PRINT"SQQQQQQQQQQQQQQQQQQQQQ           ";VS$;
1360 PRINT"SQQQQQQQQQQQQQQQQQQQQQPHRASE";:INPUT VS$:IF LEN(VS$)>30 GOTO 1350
1370 VX$(VN,K)=VS$:GOSUB1680:GOTO1290
1380 RETURN
1390 REM *
1400 REM *** PLAY MODE ***
1410 REM *
1420 PRINT"SQQQQQQQQQQQQQQQPLAY MODE (A-Z) RRETURNTO EXIT          "
1430 GET K$:IF K$=""GOTO 1430
1440 IF ASC(K$)=13 THEN RETURN
1450 K=ASC(K$)-64
1460 IF K>26 OR K<0 GOTO 1430
1470 SP$=VX$(VN,K)
1480 PRINT"SQQQQQQQQQQQQQQQQQ";K$,SP$
1490 GOSUB 1820:REM OUTPUT PHONEMES
1500 PRINT"SQQQQQQQQQQQQQQQQQ                                       ";
1510 GOTO 1430
1520 REM *
1530 REM *** LOAD WORDS FROM TAPE ***
1540 REM *
1550 PRINT"SQQQQQQQQQQQQQQQQLOAD WORDS FROM TAPE";
1560 OPEN 1,1,0,"WORDS"
1570 FOR I=0TO9:FOR J=1TO26:INPUT#1,VX$(I,J):PRINTI;J;VX$(I,J);" ";:NEXTJ,I
1580 CLOSE 1,1,0
1590 RETURN
1600 REM *
1610 REM *** SAVE WORDS ON TAPE ***
1620 REM *
1630 PRINT"SQQQQQQQQQQQQQQQQSAVE WORDS ON TAPE                        ";
1640 OPEN 1,1,2,"WORDS"
1650 FORI=0TO9:FOR J=1TO26:PRINT#1,VX$(I,J);CHR$(4):PRINTI;J;VX$(I,J);:NEXTJ,I
1660 CLOSE 1,1,2
1670 RETURN
1680 REM *
1690 REM *** PRINT OUT WORD SET ***
1700 REM *
1710 PRINT"CS";
1720 FOR X=1TO26:PRINT TAB(2)CHR$(X+64);" ";VX$(VN,X);"  ";
1730 NEXT
1740 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQWORDSET#";STR$(VN)
1750 RETURN
1760 REM *
1770 REM *** SELECT VOCABULARY SET ***
1780 REM *
1790 PRINT"SQQQQQQQQQQQQQQQVOCAB SET # (0-9)";
1800 INPUT VN:IF VN<0 OR VN>9 GOTO 1790
1810 RETURN
1820 REM *
1830 REM *** STRING HANDLER ***
```

```
1840 REM *
1850 IN=0:REM DEFAULT INFLECTION
1860 FOR P=1TOLEN(SP$):S$=MID$(SP$,P,1):IFS$<>""THEN PH=ASC(S$)
1870 IFS$="π"THENI$=MID$(SP$,P+1,1):IFI$<>""THENIN=VAL(I$)*64:P=P+1:GOTO1890
1880 GOSUB 2010:REM OUTPUT PHONEME
1890 NEXT P
1900 GOSUB 2210:REM STOP OUTPUT
1910 RETURN
1920 REM *
1930 REM *** SET 4010 ADDRESSES ***
1940 REM *
1950 DP=59471:REM DATA PORT ADD.$E84F
1960 CP=59468:REM CNTRL PORT ADD$E84C
1970 RY=59469:REM RDY SGNL PORT $E84D
1980 PA=59457:REM CLR RDY SIGNAL $E841
1990 DD=59459:REM DATA DIR REG $E843
2000 RETURN
2010 REM *
2020 REM *** OUTPUT PHONEMES ***
2030 REM *
2040 REM PH = PHOMENE CHARACTER VALUE (0-63):IN = INFLECTION (0-3)
2050 ZZ = PEEK(PA):REM CLEAR READY SIGNAL
2060 POKE DP,(PH AND 63)+(IN):REM WRITE DATA BYTE TO 4010
2070 ZZ= PEEK(CP):REM READ CONTROL PORT TO SEND DATA READY TO 4010
2080 POKE CP,ZZ AND 223:REM CLEAR BIT 5
2090 POKE CP,ZZ OR 32:REM SET BIT 5
2100 ZZ = TI + 300:REM SET MAX WAIT TIME
2110 IF ZZ<TI THEN PRINT "4010 NOT WORKING":STOP:REM STOP PROGRAM
2120 IF (PEEK(RY)AND 2) = 0 THEN 2110:REM WAIT FOR READY SIGNAL
2130 RETURN
2140 REM *
2150 REM *** 4010 STARTUP ***
2160 REM *
2170 POKE DD,255:REM SET DATA DIR REG TO ALL ONES
2180 POKE CP,237:REM SET CONTROL & UPPER CASE SCREEN
2190 GOSUB 2210:REM SEND STOP TO 4010
2200 RETURN
2210 REM *
2220 REM *** STOP OUTPUT ***
2230 REM *
2240 REM
2250 PH = ASC("-"):GOSUB 2010:REM SEND STOP TO 4010
2260 RETURN■
```

JINSAM

Data Manager selected by NASA, Kennedy Space Center

SAVE TIME! SAVE MONEY!
JINSAM saves taxpayers 100's of thousands of $

OFFICIALLY APPROVED BY COMMODORE

"Much more powerful than you can imagine!"
Robert Baker
Kilobaud Microcomputing

JINSAM is an integrated system. It makes it easy to use your information to its fullest. No more will hundreds of valuable hours be spent searching or analyzing needed information nor re-entering information for various reports.

JINSAM transforms your desk-top computer into the "state of the art" data processing machine with features and accessories found nowhere, even at 10 times the price. NASA, Kennedy Space Center selected JINSAM 8.0 and saved approximately $95,000 over other software/hardware costs. Riley County, Kansas also selected JINSAM 8.0 and saved approximately $90,000 over other software/hardware costs.

JINSAM is designed for you. It is forgiving. It has help commands for every option, available at the touch of a button. The amount of information you store, its structure and/or your hardware can change but your data won't have to be re-entered. Recovery utilites are included even for catastrophes, security passwords are built in for privacy, simple editing and entry includes auto recall, and deleting records is easy and the space is reclaimed. JINSAM includes TWO FREE accessories for reports and labels. You have unlimited report formats with summing and lined up decimals and the label printer prints up to 5 across – any size and even allows single envelopes or index cards.

JINSAM 1.0 allows fast and easy file handling, manipulation and report generation for any CBM computer with CBM 2040 disk drive. It features a menu for ease, has encrypted passwords, 3 deep sorts, .5 to 3 second recall.

JINSAM 4.0 for CBM 4000 series adds JINSORT, user accessible machine sort of 1000 records in 15 seconds; compaction/expansion of information, automatic list maintainance, unlimited number of fields, unlimited record length and much more.

JINSAM 8.0 for CBM 8000 series has all 4.0 features plus unlimited sort, horizontal format, and search by key or record number.

JINSAM 8.2, NEW FOR '82 expands 8.0 capabilities by adding information search by word, key or record number and machine language print, format and manipulation routines.

★CUSTOM DATA FILES
★CUSTOM REPORTS/LABELS
★KEYED RANDOM ACCESS
★FAST/EASY/MENU DRIVEN
★MULTIPLE SEARCH KEYS
★PRIVACY ACCESS CODES
★WILD CARD SEARCH

JINSAM EXECUTIVE version (soon to be released) is our most powerful professional system for the CBM 8000 and 9000 series. Executive will have 8.2 extended features plus allow multiple users with in-use lockout protection, executive command files, automatic math relations, join, merge or link files, greatly increased record capacity and machine information search by word, as well as by key or record number and many, many more features.

There are currently 7 more interfacing modules – and more under development, including independent interfaces between JINSAM and business packages for your G/L, A/R, A/P needs. We announce the availability of modules and enhancements in JINSAM's quarterly newsletter.

WORDPROPACK - Intelligent interface for WordPro 3, 3+, 4, 4+, creates lists of information from JINSAM files. It allows up to 10 conditions based on each item of information. Produce individualized letters, report cards, special reports, checks, invoices, etc.

MULTI-LABEL - Prints multiple labels per record with up to 2 lines for messages and consecutive numbering. Produce inventory, caution labels, bulk mail labels, etc.

MATHPACK - global calculator/editor +, −, *, /, by another field or constant; null (remove contents) of a field or replace contents of a field with any word, number or phrase. Sum multiple fields in each record or running sum of single field in all records. Extract information or effect permanent change. Replace in the same field or place in a waiting field.

DESCRIPTIVE STATPACK - Determine MEAN, MEDIAN, MODE, STANDARD DEVIATION, VARIANCE, RANGE. Generate HISTOGRAMS from 1 to 25 steps, and produce Z-SCORE reports.

ADVANCED STATPACK - (You must also acquire DESCRIPTIVE STATPACK) Generate one, two or three way CROSSTABS (number of occurance) CHI SQUARE, LINEAR REGRESSION, with graphic representation and prediction, LINEAR CORRELATION and SIMPLE ANALYSIS OF VARIANCE.

CALCPACK - 2 way interface to VisiCalc or any user program. It lets you use VisiCalc for complex manipulation, editing, placing results in JINSAM for sorting, storing or moving data to WordPro as well as giving the ability for exchange with your own applications.

INTERAC - Interface which can read VisiCalc files, WordPro files and almost any sequential files to build JINSAM databases automatically. For example: You could "download" information on holerith cards to sequential files and INTERAC would place them into JINSAM files.

All accessories are accessed thru the JINSAM menu and require security password to gain entrance.

JINSAM gives you FREEDOM OF CHOICE. Start with JINSAM 1.0 and upgrade hardware and data at any time. Choose from accessories at any time. The JINSAM Newsletter brings the latest updates, user input and uses and keeps an eye on the future.

JINSAM stands alone by placing "a lot of potential computing power in one integrated program package" (Fred Klein, Ferson div. of Bausch & Lomb). "The JINSAM package is justification for buying a system no matter what the hardware, be it Vector or Commodore or whatever the system" (Larry Colvin, Micro Computer Systems). It is a "grandfather" in this young field since the first JINI MICRO database system has been marketed since 1979. JINSAM EXECUTIVE will be the third generation in development. All JINSAM systems are sophisticated and flexible yet easy to use. JINSAM is saving its users valuable time and money in government, educational and research institutions, business and industry worldwide.

JINSAM is a Commodore approved product. See your local dealer for a demonstration.

JINSAM Data Manager

.... Additional Information and nearest dealer
.... Newsletter Subscription ($5 US/$8 Worldwide)
.... JINSAM Demo Disk ($15, plus shipping & tax)
.... User's Guide ($40.00, plus shipping & tax)

Please send to:

Name _____

Position _____

Company _____

Address _____

City, State, Zip _____

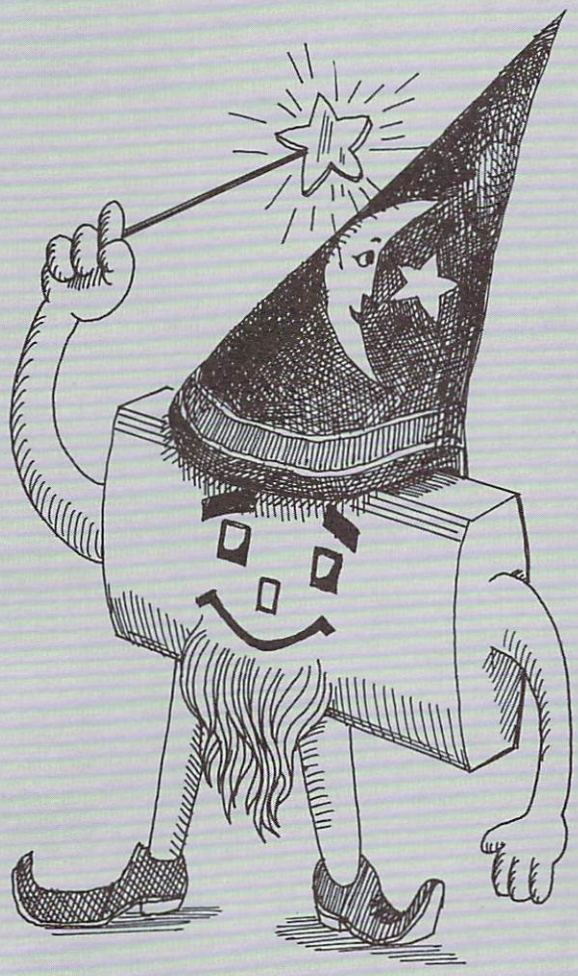
Phone ( ) _____

Computer, Disk _____

Application _____

JINI MICRO-SYSTEMS, Inc.
Box 274-CI • Kingsbridge Station
Riverdale, New York 10463
(212) 796-6200

Dealer Inquiry Welcome



The VIC Magician
The VIC is a Super Calculator

by Michael S. Tomczyk

Like any computer, the VIC is a super calculator . . . and you don't need a special cartridge or add-on to make the VIC "Calculate!"

VICBASIC, the language of the VIC, has a full mathematical operating system built into it. The special symbols which you use to perform mathematical calculations are called *arithmetic operators*. These operators are discussed in the VIC owner's guide, and also in the new VIC PROGRAMMER'S GUIDE ($16.95, available from your Commodor dealer).

VIC operators include the following symbols:

- − minus sign (subtraction)
- + addition sign (addition)
- / fraction or division sign (division)
- * asterisk (miltiplication)
- < less than
- > greater than
- = equal sign
- ↑ exponentiation

These symbols are located conveniently on the keyboard so you can use them quickly and easily. For example, the +, −, *, ?, ↑ and = signs can all be typed in without having to SHIFT. They are grouped on the right side of the keyboard.

Before going any farther, please note that the VIC does not accept commas as part of a number. In other words, always write 34000 instead of 34,000. If you include a comma you will either get an error, or the VIC will interpret the comma as a TAB command and separate the number 34,000 into two numbers, 34 and 0.

Calculating in Direct Mode

Let's take a brief look at how some of these arithmetic operators work. To begin with, we can work in the "direct" or "immediate" mode to perform calculations. We do this by using the PRINT statement. Try this:

PRINT6+4 (and hit RETURN)

The VIC responds by immediately performing the calculation . . . and PRINTs the answer. . . 10. Note that you DON'T USE QUOTATION MARKS when using the PRINT statement to do calculations. Try these two examples to see why:

PRINT"6+4" (hit RETURN) The VIC displays 6+4 instead of 10.

PRINT"The sum of 6+4 is"6+4 (RETURN). VIC CALCULATES Numbers OUTSIDE the quotes, but PRINTs the numbers and words INSIDE the quotes. This is a useful technique to remember.

Mathematical Calculation—Examples & Notes

Addition

PRINT500+1000

Subtraction

PRINT6−4

Multiplication

PRINT5*5

Computer multiplication uses the * (asterisk) instead of the X (times) sign to avoid confusion between the multiplication sign, the letter x, and the graphic symbol X. Just remember that when you are using your VIC to multiply numbers, the asterisk (*) is used instead of the (X) times sign.

Division

PRINT10/5

Division on the VIC uses the fraction symbol (/) instead of the division sign (÷). To divide 10 by 5, type 10/5 instead of the conventional 10÷5. The VIC gives you the answer 2.

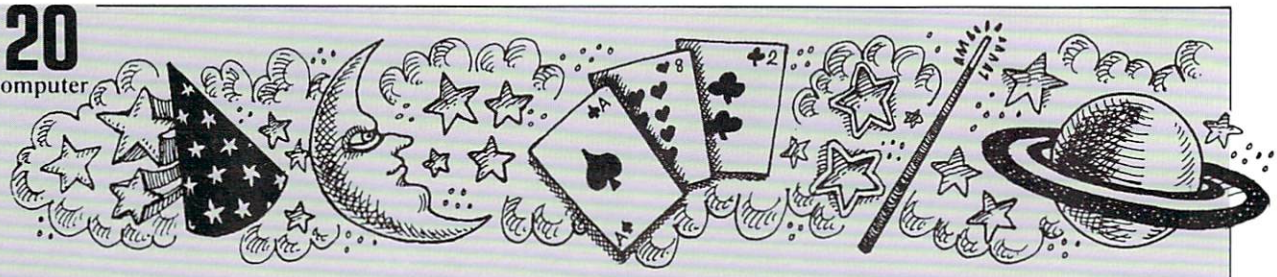
Fractions

PRINT510*10/5

Fraction are handled like normal fractions. 5/10 multiplied times 10/5 equals the whole number 1 and that's the answer the VIC gives if you type in this example.

Decimals

→



```
PRINT5.2/5.8
```

The answer to 5.2 divided by 5.8 is .896551724. The VIC handles decimal answers in the range from 0.01 to 999,999,999 in standard notation, but numbers beyond this range are automatically converted to *scientific notation*. This means that if a number is too large to conveniently display or work with, it is converted to a shorter "scientific" form (See below).

Negative Numbers

```
PRINT-5*10    . . . or . . .  PRINT-5*-10
```

The VIC recognizes and handles negative numbers, but complex formulas may require parentheses to keep negative numbers from being confused with subtraction operations. For example, −5 might be typed as (−5) if these are many operations in a single calculation.

Exponents

```
PRINT2 ↑ 2
```

This means PRINT 2 to the 2nd power, normally written as 2^2 (or 2 times 2). The VIC uses the up-arrow to show this. 10 to the 3rd power (10 times 10 times 10) is written in VIC terms as 10 ↑ 3 . . . (1000). If you're not familiar with exponentiation, 10 ↑ 3 means takes the first number (10) and multiply it times itself 3 times. 5 ↑ 2 means multiply 5 times 5 (or 25). Exponentiation is important because it is used by the VIC to designate numbers which are too large to express in normal form.

Using Pi

One of the VIC's special characters is the "pi" Symbol. Pi looks like this: π and is located on the **front of the up-arrow key.** You can print or display the pi symbol like any other character, but pi can also be used as a **value** like a number. The value of pi is 3.14159256 . . . etc. and represents the ratio of the circumference of the circle to its diameter. If you multiply the diameter of a circle by pi you will get the circumference. To use the VIC's pi charater, simply hold down the SHIFT key and press the pi key. Try typing the following:

```
PRINTπ
```

The VIC responds by displaying the number 3.14159265. Now let's say you wanted to calculate the circumference of a circle with a diameter measuring 5 inches. Simply type: PRINTπ*5 and hit RETURN. The VIC answers 15.7079633. Likewise, you can find the diameter of a circle by dividing a circumference by pi.

The Order in Which VIC Calculates

If you are using a complex formula or several mathematical operations in a long calculation, the VIC will always calculate in the same order . . . according to the **type of operation.** The order in which the VIC examines a calculation or formula is: 1) exponents, 2) multiplication and division, 3) addition and subtraction. If there are several calculation in the *same category,* the VIC performs them starting **from left to right.**

This is especially important if you are using the VIC to solve equations and perform multi-operation calculation or formulas.

You can force the VIC to calculate in the order you want by using parentheses ( ) to isolate operations you want performed separately, in which case the VIC calculates starting with the operation in the **innermost parentheses.**

Let's look at an example. Type in the following and hit RETURN:

```
PRINT10*2 ↑ 2/4+3-2
```

In this formula, the VIC will perform the calculation in the following order: first the exponentiation ( ) is considered. Next comes multiplication and division, followed by addition and subtraction. If there are several operations of the same type, they are performed in order from left to right. We can use parentheses to show you better how the VIC actually sets up this calculation. The VIC performs the calculations in order as shown and displays 11 as the answer.

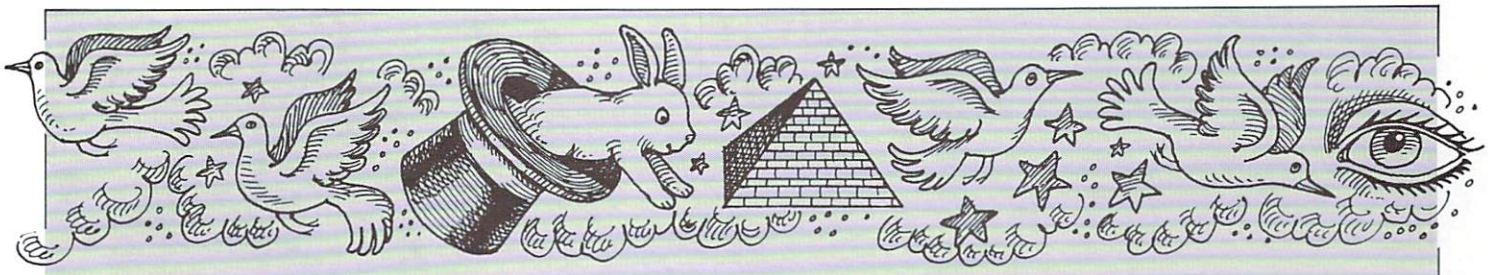
Calculation Trail	Specific Operation Each Time
10*(2 ↑ 2)/4+3−2	2 2 = 4
(10*4)/4+3−2	10*4 = 40
(40/4)+3−2	40/4 = 10
(10+3)−2	10+3 = 13
(13−2)	13−2 = 11

You could **change** this calculation by adding parentheses, like this:

```
PRINT10*2 ↑ 2/(4+3-2)
```

In this case, the VIC will first perform the exponentiation, then the multiplication . . . but before dividing, it will do the parenthetical operation (1+5−3). The calculation trial now looks like this:

10*2 ↑ 2/(4+3−2)	(4+3−2)= 5
10*2 ↑ 2/5	2 ↑ 2= 4
(10*4)/5	10*4= 40
40/5	40/5= 8



Using Numbers in Programs

This part of out "magical" VIC tour is going to show you how to use numbers in your programs.

You can use numbers in your BASIC programs—not only directly, like you'd use numbers in counting or calculating—but also **indirectly** in the form of **variables** that can change as a number changes.

Our previous section gave you a quick introduction to calculation on the VIC. We mostly worked in the DIRECT MODE. Our next step is to explore how to use numbers in your BASIC PROGRAMS.

Let's begin with a short example.

```
10 INPUTX          (and hit RETURN)
20 PRINTX*10       (and hit RETURN)
30 GOTO10          (and hit RETURN)
```

Type RUN to start the program. Now type **any number** and hit RETURN. The VIC multiplies you number by 10 and gives you the result. If you wanted to multiply your number by 10 percent, or by pi or by any other number, all you have to do is change the 10 in line 20 to something else. Also, the calculation in line 20 could also be division, addition, subtraction, or any other calculation.

The key to this program is the INPUT statement in line 10, which acceptes the number *YOU* type in, and assigns it the value X. After it multiplies your number times 10 and PRINTs the result, the program loops back and asks for another number, which becomes the "new X."

To really understand how this program works . . . and in fact how most number oriented programs work . . . you should understand **numeric variables.**

Numeric Variables

The concept of variables is explained in the VIC user manual, but this is one of the hardest computing concepts to grasp so we're going to talk you through it gradually.

A variable is like a code. Numeric variables are like codes which the VIC uses — and which you can use—to stand for a number. Numeric variables help the VIC remember and manipulate numbers—even change them—in a program. You can use variables for words, letters, phrases and graphics, too, but in this discussion we'll concentrate on those variables we use to represent numbers.

There are LEGAL and ILLEGAL variable names. Numeric variable names can be a single letter, two letters, or a letter and a number. Examples are: A, X, AA, AB, P1, R4, AB2, MU5. A special kind of numeric variable, which limits the value to an integer (whole number, no fractions or decimals) has a % sign as the last character (Examples: A%, AB%, P1%, R4%).

The other type of variable is called a "string variables" and numbers used like words instead of **values** (i.e. a zip code or phone number). String variables end in a $ sign (Examples: A$, AB$) and are mostly used to specify letters, words, phrases, graphic symbols and descriptive numbers. If you tell the VIC that X=19406 then the VIC interprets the X as the **value** 19,406. But if you want that number to be descriptive, like a zip code, then you tell the VIC that X$=19406 and the number 19406 will be used like a word or phrase instead of a value.

How Variables Work

If I say X=10 then I have just created a **variable** and from now on the letter X stands for the number 10. You might ask why we don't just use the number 10 but in a moment we'll show you why using an X gives the VIC more calculating power and program flexibility.

If I tell the VIC X=10, then any time I use the variable X in my program, the VIC will think of it as the number 10. If I tell the VIC to PRINTX, the VIC PRINTs the number 10. Let's test this assumption. Type this:

```
NEW      (and hit RETURN—to erase previous program)
X=10     (and hit RETURN)
```

We've told the VIC that the variable X equals 10. We can do this in DIRECT (IMMEDIATE) mode, without using line numbers. The VIC will automatically store these variables in its memory . . . although the variables may be changed or erased if we RUN a program. Let's continue. Type this:

```
PRINTX     (and hit RETURN)
```

The VIC responds by displaying the number 10. That's because we created the X variable by saying X=10. We can create other variables, too. Type this:

```
Y=2        (and hit RETURN)
```

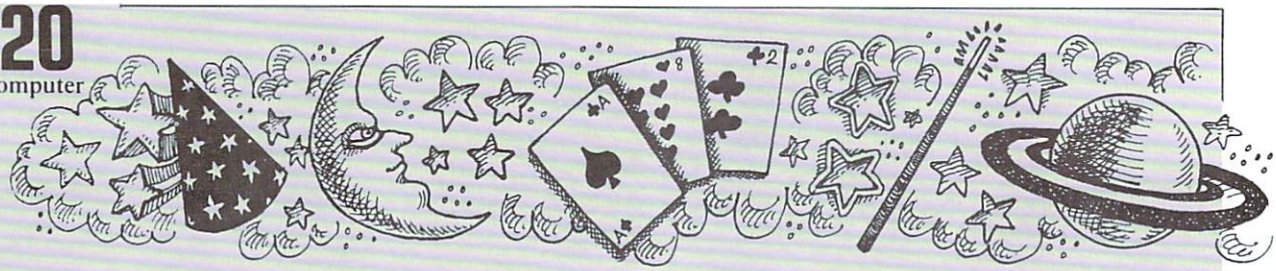
Now we've defined TWO VARIABLES. X=10 and Y=2. The **power** of these variables is easily demonstrated. Type this:

```
PRINTX*Y (and hit RETURN)
```

The VIC multiplies the value of X (10) times the value of Y (2) and displays the answer, which is 20. In addition to direct calculation, you can design all sorts of calculator programs using numeric variables. Here's a short program that lets you enter two numbers to be multiplied and gives the answer:

```
10INPUTX:INPUTY:PRINTX*Y:GOTO10
```

→



Delay Loops Use Numeric Variables

We've discussed **time delay loops** in pervious articles but how exactly does a "delay loop" work? It all has to do with numeric variables.

If you're programming a delay into your program, you may not know it but you're actually using a numeric variable. You see, you can specify that X equals a **range** of numbers—instead of a single number like 10. In a delay loop, you specify a variable like X as a range . . . the following example illustrates how:

```
10 PRINTCHR$(181);          (and hit RETURN)
20 FORX—1TO200:NEXT         (and hit RETURN)
30 GOTO10                   (and hit RETURN)
```

Type RUN and the screen slowly fills up with graphic bars. Want the bars to move faster? Hold down the RUN/STOP key and press RESTORE. Now type LIST and hit RETURN. In line 20, change the number 200 to 50 and RUN the program again. The speed with which the bars are printed picks up considerably.

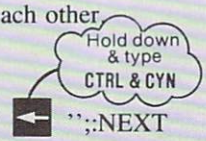
The **reason** a delay loop works is because we have defined the variable X as a **range** of numbers from 1 to 200. We then told the VIC to start **counting** from 1 to 200 before going to the next line. When we change the number 200 to 50, we shorten the loop by redefining X as a range of numbers from 1 to 50. The VIC counts faster, and the program moves faster.

In other words, we have a **loop** which PRINTs a character (CHR$(181) is the same as "["), counts to 200, then goes back around to line 10 and prints the character again. The semicolon in line 10 makes the VIC display the next character immediately next to the previous one—if we didn't have the semicolon all the characters would appear in a vertical column.

Defining a variable as a **range** of numbers has other uses, too. Say we want to PRINT a title and four lines of blank spaces at the top of the screen. That makes 88 spaces, right? (4x22 columns) Our program should tell the VIC to:

1. Clear the screen
2. PRINT a title at the top of the screen
3. PRINT 88 cyan spaces next to each other

Here's the program which does this:




```
10 PRINT" CYAN SPACES"
20 FORX=1TO88STEP1:PRINT" ";:NEXT
```

The secret is the use of the loop in line 20. First we set up a number range of X = 1 to 80. Then we STEP one at a time through that range. Each time we STEP we PRINT a cyan space. The semicolon says to print the spaces immediately next to each other. The NEXT tells the VIC to keep going through the loop until it hits the 88th time (end of the range we created for the variable X).

Here's a similar example using SOUND:

```
10 X=36876:POKE36878,15
20 FORS=128TO200STEP2:POKEX,S:NEXT
```

Here we use two variables instead of one. X is our sound speaker, V is volume. The *range* of X is the range of tone values we want to use (see table of musical notes in owner's manual). We are STEPping 2 at a time to achieve a faster effect. What actually happens is that the VIC POKEs 36876,140 . . . then POKEs 36876,142 . . . then 36878,144 . . . and so on until we reach the upper limit of the range, which is 200. We could just as easily reverse the range and STEP −2 from 200 to 140.

To make this progression of musical notes into a **sound effect,** we just speed up how fast we STEP through the range of notes. Do this by LISTing the program and changing the STEP2 in line 20 to STEP10. Another alternative is to shorten the **range** of note vaules. In line 20, try keeping the STEP2 but change the low value from 140 to 170. The higher value makes a higher sound. The shorter range means fewer notes, and a faster effect.

Using X and X+1

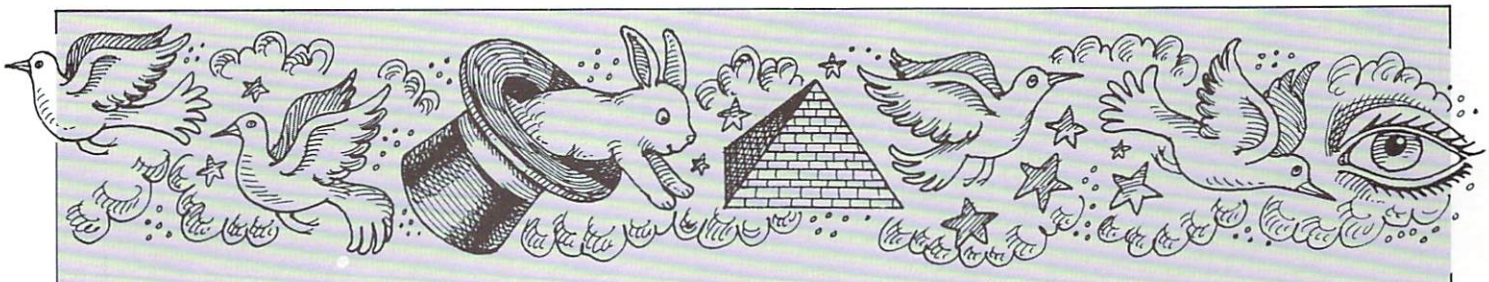
One of the most frequent uses of numeric variables involves defining a variable at the beginning of a program, then **changing** the value of that variable later on. Often this takes the form of defining a variable like X, then adding 1 to that value to increase it (X=X+1).

The following counting program illustrates the use of a variable X and X+1, as explained in the comments below:

Program	Comments
10 X=1	Define X as the value 1.
20 PRINTX	Print the value of X, which is 1.
30 X=X+1	Increase the value of X by 1 (now 2).
40 GOTO20	Go back to PRINT the NEW VALUE OF X (2).

The counting program defines X as 1, PRINTS the starting value (1), then **changes** the value X by adding 1, and PRINTS the new value, then loops back each time to increase X by one, and PRINT the new value, etc.

If you want to see this program work more slowly, try adding a delay loop by changing line 40 to read: 40 GOTO20:FORT =1TO100:NEXT



Delay loops like this can often be added anywhere in a program, by adding a colon and the FOR . . . NEXT loop.

Using the DEF FN Statement

In addition to its standard calculation abilities, the VIC also has a very special BASIC statment called "DEFINE FUNCTION," which lets you define a predetermined mathematical or scientific formula, and plug numbers into that formula during your program.

It helps to know something about variables before using the DEF FN, but this is not absolutely necessary. Pay close attention to the examples and experiment with a few of your own and you should catch on quickly to how this powerful mathematical statement is used.

Let's begin with a simple example. Type NEW and hit RETURN to erase any old programs and type in this program:

Program		Explanation
10 DEFFNA(X)=.10*(X)	(RETURN)	Sets up the formula.
20 INPUTX	(RETURN)	Sets up your number input.
30 PRINTFNA(X)	(RETURN)	Prints the formula answer.
40 GOTO10	(RETURN)	Goes back to repeat line 10.

Type RUN and hit RETURN to start. (Hold down RUN/STOP and RESTORE to exit the program). This program calculates 10 percent of any number you type. When the question mark appears, type a number and hit RETURN. The VIC calculates 10 percent of that number and diaplays the answer.

Setting up a DEF FN Statement

Let's use our example to show you how the DEF FN statement works. To begin with, the format is always similar:

DEFFNA(X)=(formula)(X)

The DEFFN part of the line is simply the DEFine FuNction statement.

The A following the DEFFN statement is a **variable.** It means that this function has been defined (by you) as Function A. It might be helpful to think of this as the name of the particular formula you are using. The **name** you give to the formula must always be included where the **A** is shown. Most programmers save memory by using ONE-LETTER VARIABLES, but the function name can be any legal numeric variable name, up to **FIVE CHARACTERS LONG.** The name must start with a letter but can also include numbers. Here are some function name examples:

Legal DEFFN Names:		Illegal DEFFN Names:	
DEFFNP10	DEFFNWAR2	DEFFN1	(number)
DEFFNAB	DEFFNGAME	DEFFN82M	(starts w. number)
DEFFNABCDE	DEFFNR2D2	DEFFNABCDEF	(too long)

The DEFFNA is the name of the DEFINE FUNCTION statement. The variable in the parentheses (X) is the NUMBER we are going to manipulate or include in our calculation. DEFFNA(X) simply means we are defining a function called A which is going to manipulate a number called X. Using an X is just a formality. Even if use use X here, Later on, you can put ANY NUMERIC VARIABLE in the formula by referencing FNA(B2), FNA(B+C+D), FNA(XY) or any other variables in your program. (see ADDING MACHINE with ROUNDING, BELOW). You aren't limited to using the X variable. In other words if you use X or Y or B2 or whatever in your DEFFN formula setup, you can plug other variables in the formula by using FNA (your other variable here), because the A in the FNA activates the formula.

The next part of our DEFFN example is the **equal sign.** This means we are setting FUNCTION A equal to a formula which we are going to type on the RIGHT SIDE OF THE EQUAL SIGN. This formula will control what the function actually does.

The right side of the equal sign consists of two parts . . . the **formula or calculation** we want to perform, and the **place** where a number can be inserted. The number is represented by the variable X in our first example and sample format, but it can be any **legal numeric variable name** (a letter, 2 letters, a letter and a number or 2 letters and a number). Where we put our number in the formula (where we put our X) is important because this number will be the KEY to our calculation. In the first example—out 10 percent solution—we made the formula very simple. We multiplied X times 10 percent. Now, every time the VIC encounters an X we can tell it to multiply it times 10 percent simply by including FNA(X) in our program.

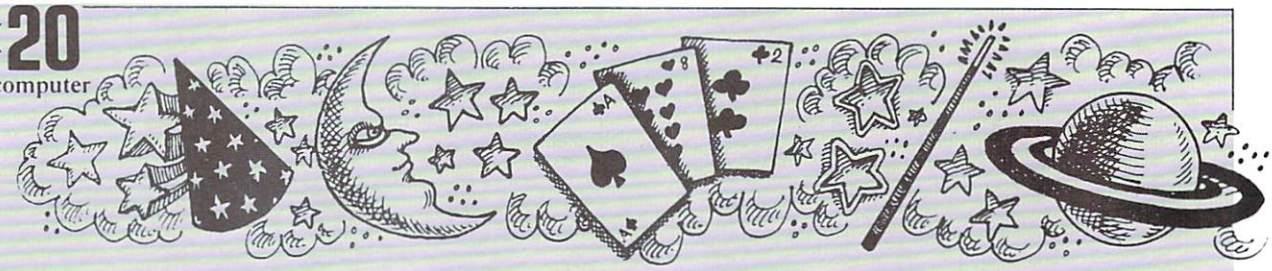
Celsius TO Farenheit Using DEF FN

Here's an example of a DEF FN formula which converts farenheit into celsius . . . we'll call the function C (for Celsius) and call the Farenheit number in the formula F. The function itself looks like this:

DEFFNC(F)=5*(F−32)/9

Celsius temperature equals 5 times the (Farenheit temperature minus 32 degrees) divided by 9. We've set up a function statement which does just that. Now all we have to add to our program is a means for us to ENTER the Farenheit number. This is very similar to our 10-percent program above:

```
10 DEFFNC(F)=5*(F−32)/9
20 PRINT"ENTER FARENHEIT":INPUTF
30 PRINT"FARENHEIT"F"IS"FNC(F)
     "DEGREES CELSIUS"
```



The structure is simple. First we set up our formula, a straightforward temperature conversion. We call our statement C and the number we want to convert is F for Farenheit.

Next, we PRINT a message telling the user to type an input.

The INPUT F tells the VIC will display a question mark and wait for the user to type (input) a number. As soon as the user types a number, it is automatically given the variable code name "F" (see previous section).

Finally, we PRINT a message which includes the calculation answer. We do this by PRINTing a verbal message INSIDE the quotation marks, then we PRINT the value of F (the number the user typed in from the keyboard) OUTSIDE the quotation marks. Next, we go INSIDE quotes again to type the word "IS", then back OUTSIDE to print the VALUE of FNC(F)—which means the value of F after plugging it into our temperature conversion function statement C in line 10. That value is PRINTed and we're back INSIDE quotes again for the remainder of the message (DEGREES CELSIUS).

You can dress up this program considerably by adding color (after any quotation mark, hold down the CTRL key and hit one of the keys with a color on the front), or REVERSE (inside quotes just like color except press CTRL and RVS ON, then CTRL and RVS OFF to get back to normal).

Here's a SOUND EFFECT you can add to this program. Just type in this line and hit RETURN, then type RUN:

```
25 POKE 36878,15:FORS=200TO125STEP-1:
     POKE36876,S:NEXT
```

This line sets the VOLUME control at 15 (highest), then creates a loop with a musical note range from 200 to 125 and steps down one at a time when the program hits POKE36876,S:NEXT. You should recall from the VIC owner's guide that 36876 is one of the VIC's speaker numbers.

A DEF FN Rounding Program

Here's a program which demonstrates the DEF FN statement, and also provides a "rounding" program. To use it, you have to use a variable (in this case X) to stand for your number. Then, whenever your number, or any number designated X, is generated, it can be **rounded** to the nearest 10th, 100th, 1000th or whatever simply by changing the value of P in line 20. Change P to 10 to round in 10ths for example. Here is the sample program:

```
10 DEFFNA(X)=INT(X*P+5)/P
20 P=100
30 PRINT"ENTER A NUMBER":INPUTX
40 PRINT"YOUR NUMBER ROUNDED TO THE
     NEAREST"P"TH IS"FNA(X)
50 GOTO10
```

A simple Adding Machine

Here's a simple program using the INPUT statement which creates a simple adding machine:

```
 5REM ADDING MACHINE
10INPUTA:PRINTC+A
20INPUTB:PRINTC+A+B
30C=A+B+C
40A=0:B=0
50GOTO10
```

Here's how this program works . . . first we INPUT a number which we call A. Then we PRINT the value of C and A, which on the first round does nothing (because C is not yet assigned a value) and only the A number is PRINTed.

In line 20 we INPUT the value of the second number, which we call B. Then we PRINT the sum of A + B (our adding machine) but again no value is assigned to C yet so the C is ignored and we get the sum of the first two numbers.

In line 30 we create a new variable C and define C as the sum of A and B . . . in other words, C now becomes the value of the first two numbers added together.

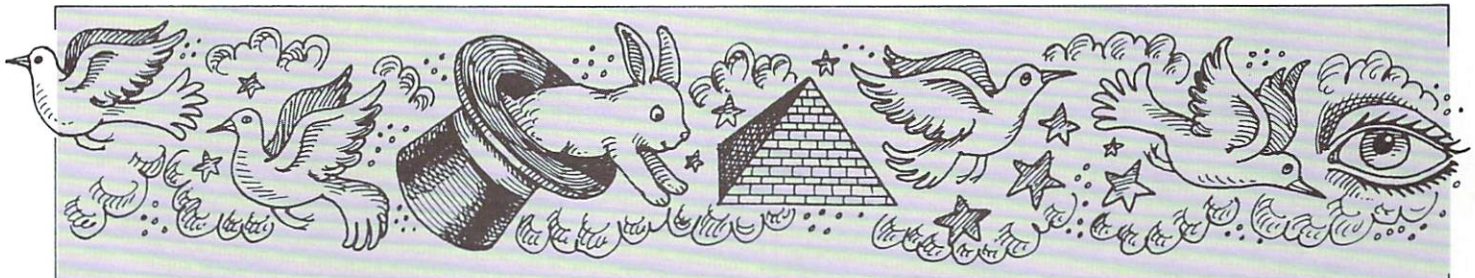
In line 40 we reset A to zero and B to zero. This is called "clearing our variables" and means here that we put A and B back to zero so we could type in NEW VALUES for both of these numbers. The sum of the two numbers we already typed in is preserved in our new variable C.

Now in line 50 we go back to line 10 and repeat the program, except NOW the C variable has a value (the sum of the first two numbers we typed in). So when we type in a new number called A (INPUT A), the VIC PRINTS the sum of A (the new number) and C (which we define as the sum of our first two numbers). The result is the added sum of the first three numbers we've entered.

In line 20 we enter the next number (INPUT B) which we add to our new number (A) and our old sum (C). The total is PRINTed.

Line 30 redefines C. This time, C becomes the total of the old sum (C), our new number A, and our new number B. Now we can set A and B back to zero and start over again.

More specifically . . . if we were adding 10 every time, we would start by adding 10 (A) to 10 (B) which then becomes 20 (A+B). C is then defined as 20 and put A and B are set back to zero. Then the program repeats to add a new number 10 (new A) to 20 (C) to get 30 . . . then adds 10 (new B) to 30 (C) to get 40, then redefine C as 40, or 10 (new A) + 10 (new B) + 20 (old C).



Adding Machine With Rounding Function

If we want to add decimal point numbers and ROUND them off, we can do so by incorporating our ROUNDING function described previously. Here's how we would add the rounding function to our ADDING MACHINE program to round off all numbers to the nearest 100th (2 decimal places rounded up or down):

```
 5REM ADDING MACHINE WITH ROUNDING
10DEFFNR(A)=INT(A*100+.5)/100
20INPUTA:PRINTFNR(C+A)
30INPUTB:PRINTFNR(C+A+B)
40C=A+B+C
50A=0:B=0
60GOTO20
```

This rounding program takes our A, B and C variables and plugs them into the DEFFN rounding formula which we described earlier. Now ANY value put in the parentheses after DEFFNR( ) will automatically be rounded and plugged into the formula, because it's the DEFFNR that determines what is done to the information in the parentheses, and after the FIRST definition of the function (line 10) any time we want to plug a variable into the formula, we can do so by typing DEFFNR followed by the variable we want to manipulate in parentheses.

In the program above, notice that we only had to type an A in parentheses in line 10. Later, when we type in line 20: FNR(C+A) the program automatically inserts the C+A variables where the A was in line 10. The C+A variables could have been any other numeric variables we might use in our program—M, R, XX, S2, etc.

Scientific Notation, Binary Decimal & Other Peculiarities

Because the VIC calculates using the binary number system, and translates it into our normal decimal numbering system, there are a few peculiarities which may arise . . . for example, if you type: PRINT.34−.30 you will get an answer which looks like this: .0400000001

Clearly, the extra decimal places and the last 1 do not belong in our answer, which should be .04. The difference is so minimal that it doesn't affect most calculations. The best way to avoid this discrepancy is to use the rounding routines discussed earlier.

Another mathematical quirk is if you type: PRINT.5555555556 The VIC will PRINT .555555555 and lose the 6. The VIC rounds DOWN automatically at six or less digits, and rounds UP if there are seven or more digits. This results from the way computers store floating point numbers.

Another idiosychrosy of the VIC is its use of **scientific notation** when an answer or calculation exceeds a certain limit. This special notation allows the VIC to display large numbers using fewer digits, and is used by many computers. Scientific notation takes the form:

numberE+ee

Here are some examples:

Standard		Scientific Notation
2E1	=	20
105000	=	10.5E+4
6600	=	66E+2
.66	=	66E−2
.0000000001	=	1E−10

Summary

If you are planning to use your VIC for calculations, these notes which supplement your user's manual should help. The VIC PROGRAMMER'S REFERENCE GUIDE provides additional information, and most BASIC programming manuals for the PET/CBM or VIC will give you more insights. ■

VIC=20
The friendly computer

VICMODEM MEANS ANYONE CAN AFFORD TELECOMPUTING

Imagine being able to dial up stock quotes, newswire articles, research information and more . . . *over the telephone*! That's what the VICMODEM lets you do. It means you can access vast quantities of sophisticated computer programs, information data banks and encyclopedia-type information, quickly and inexpensively using three small components: the VIC 20, VICMODEM, and your telephone.

The VICMODEM retails for under $110 and combined with the VIC 20 which sells for under $300, you can get into telecomputing for under $410! This price breakthrough has far-reaching ramifications not only for personal computer owners, but also for society as a whole.

The possibilities created by the advent of a low-priced computer/modem combination are mind-boggling. During 1982, Commodore will begin publicizing the VICMODEM "connection," and give you a better idea of how many unique and wonderful opportunities are available immediately using the VICMODEM, but for now here are just a few of the telephone computing services which are available to you:

STUDENTS: You've just been given a term paper assignment. The subject is "Social Security." Everyone else in your class runs to the library and gets their information from months-old articles. You go home, connect your VIC 20 and VICMODEM to your telephone, and dial up a telecomputing service. The service has a special UPI Newswire service which lets you type in ANY KEY WORD OR PHRASE. In a matter of seconds, the computer searches ALL UPI NEWSWIRE STORIES and tells you how many stories appeared during the last three days, and in what order. You type the phrase, "Social Security" and discover there are five articles. Now all you have to do is tell the computer to play back the stories . . . slowly . . . on your television screen . . . and take notes. If you want, you can have the stories *printed out* on your VIC Graphic Printer, for future reference. After reading the articles, you call up a *wordprocessing program* from the same service that gave you the UPI newswire stories. The wordprocessing program lets you type your paper using the VIC 20 keyboard, and edit the paper on your television screen as you write it. The combination of a ready reference service and a wordprocessor are both made possible by the VIC 20 and VICMODEM, connected to a telecomputing service provided by companies like COMPUSERVE™ or THE SOURCE™.

INVESTORS: Using your VIC 20 and VICMODEM, you can call up at least three different telecomputing services and get current stock quotations and company reports . . . which print out neatly in large letters on any ordinary television set or monitor, or print out on the VIC Graphic Printer. Professional investors pay thousands of dollars for sophisticated computer terminals which do essentially the same thing, but now an average investor can have the same terminal equipment for approximately $500!

BUSINESSMEN: Most telecomputing services let you enter information about your business . . . including contracts, wordprocessed letters, accounting statistics and more . . . into a mainframe computer. You start by using one of the service's business programs to enter or tabulate your information. Then for a small fee, you can store your business information in the mainframe computer, access and update it as required. The service is backed up to protect the stored information and can be called back to your computer viewing screen (a television set or monitor), printed out on paper using a VIC Graphic Printer or letter quality printer, or stored on diskettes using the VIC SINGLE DISK DRIVE.

GAME ENTHUSIASTS: The most sophisticated computer games in the world . . . including games that require large amounts of memory not available in most small computers . . . can now be played over the telephone by tying into a large mainframe computer which stores the game and performs strategic calculations. Several players—even several *dozen* players—*each in a different city*, can participate in the same game!

One example is an experimental service in Chicago which creates a computerized "house of games" which subscribers can call up on the telephone, using their computer. The subscriber "enters" the house and proceeds from room to room, playing games in each room, chatting with *other players* in the rooms . . . and much more.

A fascinating science fiction book called TRUE NAMES by Vernor Vinge describes a similar experience whereby a computer operator ties into the government's computer system through his computer and telephone, and meets other computer owners in an imaginary fantasy world which is much like the "game house" in Chicago.

The arrival of a service which only last year was science fiction shows how fast our society is moving, and how the invention of a low-cost modem can help bring the future into the present. It's interesting to note that practically *any* world or environment can be created using the computer. Players can tie into that "world" by connecting their computer to a modem and letting his television set screen show him where he is in that imaginary world. It's a short jump from a telecomputing "game house" to a simulated battlefield or other simulated environment where the "players" can conduct their simulated maneuvers without risking a life or wasting a dollar.

LONG DISTANCE COMPUTER CHATS: At least two major telecomputing services currently let a VIC owner in one city like Philadelphia, "talk" to another VIC owner in another city like Los Angeles . . . *for the cost of a local phone call, plus hourly 'connect time!'* This amazing service actually makes it cheaper to communicate over long distances than dialing direct on the telephone. More importantly, *several* subscribers can talk at the same time. Also, computer data, reports, programs and other information can be easily transferred . . . including business information such as daily sales or inventory reports.

There are many more examples of how the VIC 20 and VICMODEM are opening up the future . . . bringing airline schedules into your home . . . making encyclopedias available by telephone . . . providing up-to-the-minute sports scores and other information, but let's take a closer look at some of those specific services.

TELECOMPUTING SERVICES

There are several telecomputing services which you can subscribe to today, which offer the types of services described above. The most prominent microcomputer services are listed below. Typically, there is a one-time, lifetime subscription fee which may run from $9 to $100. After that, you pay only for the actual "connect time" to the service. Connect time rates range from as low as $2.75 from midnight to 6 a.m. to as high as $17 during daytime "prime time" hours. An intermediate rate of approximately $5 an hour is normal for 6 p.m. to midnight, the time when most home users access the network. ■

Light Pen Drawing Program

Would you like to draw *right on the screen of your TV,* using a "pen" controlled by your VIC 20? All you need is the simple BASIC program listed below, the Super Expander cartridge, and a light pen.

First a word about how the VIC 20 works with a light pen. The picture on your TV set is formed by an electron beam that sweeps across and down the screen at the rate of 30 pictures every second. The 6560 (V.I.C.) chip inside the computer creates the signal that controls the picture, so it "knows" where the beam is at any time.

A light pen contains a light-sensitive component that tells the 6560 chip when the electron beam is sweeping past. The electron beam hitting the light pen activates the 6560 chip, which records the horizontal and vertical position on the screen in two memory locations.

The RPEN(0) and RPEN(1) functions from the Super Expander read the horizontal and vertical values of the light pen. Experimenting showed the limits to the numbers' both directions, when the screen is set to a graphic mode. The RPEN(0) value ranged from 34 to 114, and RPEN(1) went from 28 to 108.

	RPEN(0)	RPEN(1)
TOP RIGHT	34	28
TOP LEFT	114	28
BOTTOM RIGHT	34	108
BOTTOM LEFT	114	108

The resolution of the screen with a light pen is 80 positions, both from top to bottom asnd from left to right. Since the Super Expander works on a scale of 0 to 1023, the number read from the light pen has the lower limit subtracted, which gives a range of values from 0 to 80. This number is multiplied by 12.8, which expands the range to 0 to 1024.

The light pen used included a switch that detects when the light pen is pressed down. This switch is read through RJOY(0), which will be 1 when the pen is down and 0 when it is up. This is needed because the pen is quite sensitive, able to detect light from a foot away.

This program uses the VIC 20's multi-color graphic mode. Points on the screen can take on the color of the screen, border, or an auxiliary color, or the character color corresponding to that space. There is only one screen, border, and auxiliary color at any time, but there are 400 different character colors, 1 for each space on the screen. Changing the auxiliary color will change every point drawn using that color on the screen, while changing the character color only effects the spaces on the screen that have lines drawn on them after the change. Character color should be selected when placing a color that is localized on an area of the screen.

Certain keys on the keyboard control color selection and special function, as shown in the following table:

Keyboard Controls

Color register selection:

0	screen color
1	border color
2	character color
3	auxiliary color

Change color in register:

CTRL 1	change color to black
CTRL 2	change color to white
CTRL 3	change color to red
CTRL 4	change color to cyan
CTRL 5	change color to purple
CTRL 6	change color to green
CTRL 7	change color to blue
CTRL 8	change color to yellow

Change color in register (auxiliary or screen color only):

Q	change color to orange
W	change color to light orange
E	change color to pink
R	change color to light cyan
T	change color to light purple
Y	change color to light green
U	change color to light blue
I	change color to light yellow

Special functions:

SHIFT CLR	clear screen
f1	end program, erase screen (CONT will re-start)
P	paint—the next place the pen is pressed is filled with a color—an area can only be painted once

TABLE OF FUNNY CHARACTERS IN LINE 150

```
"■      CTRL 1
"■      CTRL 2
"■      CTRL 3
"■      CTRL 4
"■      CTRL 5
"■      CTRL 6
"■      CTRL 7
"■      CTRL 8
"■      FUNCTION 1
```

LIGHT PEN DRAWING PROGRAM

```
10 GRAPHIC 1
20 DIM C(3)
25 FOR L=0 TO 3
26 C(L)=RCOLR(L)
27 NEXT
30 X0=RPEN(0)
40 Y0=RPEN(1)
60 IFRJOY(0)<>1THENR=0:GOTO120
65 X=RPEN(0)
70 Y=RPEN(1)
85 IF(X<34)OR(X>114)OR(Y<28)OR(Y>108)THEN120
86 IF PO=1 THEN:PAINTC,(X-34)*12.8,(Y-28)*12.8:PO=0
88 IFR=0THEN:POINT C,(X-34)*12.8,(Y-28)*12.8:R=1
90 DRAW C TO (X-34)*12.8,(Y-28)*12.8
100 X0=X
110 Y0=Y
120 GET A$
130 IF A$="" THEN 60
140 FOR L=1TO23
150 IF A$<>MID$("0123■■■■QWERTYUI■■",L,1) THEN NEXT:GOTO 60
160 IF L<5 THEN C=L-1:GOTO 60
170 IF L<13 THEN 190
175 IFL=21THEN:SCNCLR:GOTO60
176 IFL=22THEN:GRAPHIC4:END:GRAPHIC1:GOTO60
177 IF L=23 THEN PO=1:GOTO60
180 IF (L=0) OR (L=2) THEN 60
190 C(C)=L-5
200 COLOR C(0),C(1),C(2),C(3)
210 GOTO 60

READY.■
```

— Neil Harris

Another Voice for the VIC

Normally, your VIC has 4 musical voices . . . three music registers and a white noise register . . . but by connecting a small amplifier and speaker to the USER PORT, and doing a little programming, you can get *another* musical voice.

The user port on the VIC is very similar to the user port on the PET. This makes it easy to adapt some of the PET's music methods to the VIC.

Background—Adding Sound to Older PET/CBM's
Before Commodore introduced the CBM 8032 with a build-in speaker, most PET/CBM users had to develop their own means of getting their computers to squeek, hum, whistle, and sing. They came up with the idea of using the shift register of the 6522 connected to the user port to send square waves through an external amplifier/speaker combination. The shift register could be programmed through BASIC, giving a wide variety of squeals, pops, sirens, etc.

Theory
Most music is made up of square waves of different amplitudes and frequencies. One of the functions of the 6522 chip is to generate square waves through the CB2 line. If we connect the CB2 line to a speaker, we will be able to hear the square waves generated by the VIC.

NOTE: Connecting a speaker directly to CB2 may damage your VIC and void your warranty. You must connect the speaker through an *amplifier* to protect the VIC.

Parts Needed
1. Small battery powered speaker/amplifier
2. User Port Connector (12 position, 24 contact edge connector with .145″spacing
3. Wire

Connecting The External Speaker to Your VIC

Bb	=251	(B below first C) B	=124
C	=237	(first C) C1	=117
C#	=224	C1#	=111
D	=211	D1	=104
D#	=199	D1#	= 99
E	=188	E1	= 93
F	=177	F1	= 88
F#	=167	F1#	= 83
G	=157	G1	= 78
G#	=149	G1#	= 73
A	=140	A1	= 69
A#	=132		

1. Wire the GROUND of the amplifier to the GROUND of the USER PORT (pin N) .
2. Wire the SIGNAL of the amplifier to the CB2 output of the USER PORT (pin M).

You are now ready to add your other voice through a BASIC program.

BASIC program steps:
1. Set the 6522 shift register to free running mode by typing:
   POKE 37147,16
2. Set the shift rate by typing:
   POKE 37144,C where C is an integer from 0 to 255
   C is the note to be played.
3. Load the shift register by typing:
   POKE37146,D where D = 15, 51, or 85 for a square wave. This step sets the octave for the note.

This step must be done last, since as soon as it is set, the VIC starts generating the square waves.

The frequency of the square wave can be found by the following formula:

$$\text{FREQUENCY} = \frac{500000\ \text{Hz}}{(C+2)(D1)}$$

Where D1=8 when D=15
D1=4 when D=51
D1=2 when D=85

When you're in this mode, the VIC will *not read or write to cassette*. To restore normal operations, you must type:
POKE 37147,0

The following short program demonstrates music using this method. By hitting a letter a note will be played.

```
10 PRINT " MUSIC USING CB2."
11 REM A TO G IS ONE OCTAVE, SHIFT A TO G IS
   ANOTHER
15 PRINT "HIT + TO GO UP AN OCTAVE"
17 PRINT:PRINT "USE ! TO EXIT."
20 POKE3747,16:DIMA(14):FORI=1TO14:
   READA(I):NEXT
40 GETA$IFA$=""THEN40
42 IFA$="!" THEN POKE37147,0:END:REM
   RESET 6522
45 ifa$="+" THEN SF=SF-(SF<2):GOTO40
50 ifa$="+" THEN SF=SF-(SF>0):GOTO40
60 A=ASC(A$)-64+(ASC(A$)>192)*121:IF A>14
   OR A<1 THEN 40
70 POKE 37144,A(A)
80 POKE37146,-(SF=0)*15-(SF=1)*51-(SF=2)*85
90 GOTO40
100 DATA 124,117,104,93,88,78,69
110 DATA 251,237,211,188,177,157,140
```

One use for this procedure is to connect an external amplifier and speakers to your VIC to provide improved sound quality . . . or perhaps to use your VIC as a music synthesizer, with the proper program. This is only one of several hobbyist-type projects we will be describing in the VIC section of this magazine. Watch future issues for more hobby-related computer projects. ■

— Andy Finkel

The VIC's "Hidden Symbols"

There are at least 4 special graphic symbols which are not inscribed on the VIC keyboard. These symbols are listed in the character POKE chart on page 141 of the owner's manual, but are not listed in the CHR$ chart.

The symbols are only accessible when in Upper/Lower case mode. (You get Upper/lower case by holding down the Commodore key and left SHIFT keys simultaneously).

The four hidden symbols and their CHR$ codes are shown below:

Symbol	Keytop	How Accessed	CHR$ Code	POKE Code
✓	@	SHIFT @	186 or 250	122
▒	*	COMM *	127 or 223	95
▚	£	SHIFT £	169 or 233	105
▞	↑	SHIFT ↑	126 or 222	94

The special symbols are not avilable during UPPER CASE mode, and while they are not inscribed on the keyboard, you CAN use them in UPPER/LOWER CASE mode, either by typing the proper key (from the keyboard while in lower case graphics mode). You can also use them by POKEing their values, or by using the CHR$ value.

CHR$ values are displayed in this form:

```
PRINT CHR$(186)
```

You can also use the CHR$ command in DIRECT or IMMEDIATE mode if you just want to view the special characters. The following program will display the hidden characters on the screen:

```
10 PRINT CHR#(14)
20 A=186:GOSUB70
30 A=127:GOSUB70
40 A=126:GOSUB70
50 A=169:GOSUB70
70 FORX=1TO44:PRINTCHR$(A);
     :NEXT:RETURN
```

One of these characters—the checkmark—is used in a popular new Commodore program on tape, "The VIC Typewriter." The checkmark signifies that a carriage return has been entered. Use of the checkmark on the VIC Typewriter demonstrates how the subtle use of graphics in a program which is mostly text, gives the program some extra "character." ■

VIC-Glitch

We regret that VIC-Trick #5 in the October issue of *Commodore Magazine* contained some errors that prevent the program from operating correctly. So, with our apologies, here is the correct way to DRAW A HIGH RESOLUTION CIRCLE:

VIC-TRICK #5 (DRAWING A HIGH RESOLUTION CIRCLE)

```
10  FOR S = 7168 TO 7679: POKE S,0: NEXT
20  POKE36879,8:PRINT CHR$ (147);
30  FOR S = 7680 TO 8185: POKES, 160: NEXT
40  POKE 36869,255
50  FOR L = 0 TO 7: FOR M = 0 TO 7
60  POKE 7680+M*22+L, L*8+M
70  NEXT M, L
80  FOR X = 0 TO 63
90  Y1 = 32 + SQR (64*X-X*X)
100 Y2 = 32 - SQR (64*X-X*X)
110 FOR Y = Y1 TO Y2 STEP Y2-Y1
120 CH = INT (X/8) *8 + INT (Y/8)
130 RO = (Y/8 - INT(Y/8))*8
140 BY = 7168 + 8*CH + RO
150 BS = 7-(X-INT(X/8)*8)
160 POKE BY, PEEK (BY) OR (2 ↑ BS)
170 NEXT Y,X
180 GOTO 180 ■
```

Some BASIC Analogies in Machine Code

by
Elizabeth Deal

This article presents an addressing mode conversion table between BASIC and machine language. It deals with several addressing schemes that are available on the 6502, a chip that runs the PET. The information might benefit beginners in machine language. Do not expect any revelations on a grand scale. I am one of the beginners, occasionally benefitting from seeing things from a BASIC perspective.

Relevant to the information presented here is Jim Butterfield's article in Compute #7 describing what various addressing modes can do for you and how far they "reach" in memory. An equally eloquent text is Mansfield's beginner's material (Compute, issue #10). The latter deals with machine code in general.

PEEK and POKE instructions in BASIC can be thought of as machine code instructions in that they directly look at or modify memory. For instance, POKE 32768,2 is equivalent to saying LDA #$02:STA $8000 where Accumulator contains 2 – screen code for the letter B. Conversely, Q = PEEK (32768) is just about the same as saying LDA $8000:STA in some place corresponding to Q. These are examples of immediate and absolute direct addressing modes, respectively.

Indexed addressing is no more complex than adding an offset to a known beginning address. In BASIC one might say POKE 32768 + J,2 where offset J can vary from 0 to 999 for the PET's screen. BASIC usually uses FOR-NEXT loop for this sort of work. GOTO with computed index J does the same thing. The machine code equivalent of that instruction, STA $8000,X differs from BASIC only in that the limits of X are 0 and 255. The concept is the same.

Indirect addressing is similar to BASIC's nested PEEK statements. Indirection means that instead of doing something to a location specified in an instruction, one will do something with the contents pointed by the zero page value in the instruction. The idea of pointer is equivalent to a concept of indirect addressing.

For instance, in the PET we may want to know the contents of the second byte of a BASIC program. Assume, for a moment, that you do not know that a BASIC program begins at 1025. This assumption is not as silly as one may think, because BASIC programs can be scattered throughout the PET. Unless confused by a programmer, PET always knows where the program is by keeping track of it in a pointer in locations 40 and 41. Knowing that the address of the program (or its pointer) is held there, we code V = PEEK(PEEK (40) + 256*PEEK(41) + 1).

Normally, location 40 contains 1 and location 41 contains 4. This reduces to V = PEEK(1 + 256*4 + 1) and further to V = PEEK(1026) at which point a value of V is set to whatever is in location 1026. In machine code we can write the same thing as LDY #$01:LDA ($28),Y (when Y = 0 we'll look at 1025, when it is 1, we'll look at 1026). This shows one of the two indirect addressing modes. A-register will contain the same value that V gets in the BASIC example.

The second mode is coded in machine language as LDA(ML,X). Here, depending on some condition X, we may pick values from one of several tables, stored at and following certain locations (for instance, $1000, $2000, $3000, etc.). The addresses of tables are stored in zero page, for example in low-high pairs at $4-5, $6-7, $8-9, etc. When X = 0, we'll use pointers $4-5, hence, values stored in $1000. When X = 4 we'll get to the pointer at $8-9 and pick up values from $3000. And all that is done by simply stating LDA($04,X). In BASIC this would be a long expression: A = PEEK (PEEK(4 + X) + 256*(PEEK (4 + 1 + X).

Indirect JMP (jump) instruction has no exact equivalent in BASIC. Let's invent one for a while. Imagine that a line number is held in locations 1 and 2. We might then code GOTO (PEEK(1) + 256*PEEK(2)) in BASIC. (It will not work because we renumber programs, and it will not work because BASIC interpreter will issue an error message). A SYS command can sometimes be used as if it were an indirect jump. It's a handy instruction, frequently used by the BASIC interpreter, Supermon and other such programs.

Analogies:

BASIC	MACH. CODE	EXAMPLES
*Direct and immediate addressing		
1. A = 4	LDA #V	LDA #$04
2. A = PEEK(M)	LDA M	LDA $30
3. "	"	LDA $0401
4. A = PEEK(M + X)	LDA M,X	LDA $30,X
5. "	"	LDA $0401,X
6. A = PEEK(M + Y)	LDA M,Y	LDA $0030,Y
*Indirect—note use of the "peek of peek" construct		
7. A = PEEK(PEEK(M + X) + 256*PEEK(M + 1 + X))	LDA (M,X)	LDA ($30,X)
8. A = PEEK(PEEK(M) + 256*PEEK(M + 1) + Y)	LDA (M),Y	LDA ($30),Y

Some commonly used names for those addressing modes, their "reach" into addressed memory (from Butterfield's text cited above), and the limits of parameters (positive integers only) are:

1. immediate, used for assignment, V<256
2. zero page direct, zero page, M<256
3. absolute direct, anywhere, M<65536
4. zero page indexed with X, zero page, (M+X)<256
5. absolute indexed with X, anywhere, (M+X)<65536
6. absolute indexed with Y, anywhere, (M+Y)<65536
7. indexed indirect or indirect preindexed with X, anywhere, (M+X)<255
8. indirect indexed or indirect postindexed with Y, anywhere, M<255

The names of the last two options appear unreadable and intimidating. But precise wording, in spite of its obscurity, has precise meaning, which imprecisely goes like this:

Example 7 means that at execution time the offset X is first added to an address to obtain one of several possible addresses to be used as a pointer, followed by access to the addressed value.

Example 8 means that at execution time the Y index is added to an address after the indirection in order to access one of several values in a table addressed by one pointer.

Note the position of X and Y in the BASIC analogies. X is added to M before the destination address is computed, while Y is added after the entire address has been computed.

Elizabeth Deal is a Malvern, Pa.-based free-lance writer whose work frequently appears in COMPUTE! ■

Collect

One disk command that doesn't get nearly as much attention as it should is COLLECT. BASIC 2 users will know this as the disk Verify or Validate command.

Collect causes the disk to throw away the old BAM (Block Availability Map) and rebuild a new one. The process starts with the first directory entry. The disk picks up the track and sector coordinates of the first block of the first file, and begins tracing the block chain. During the trace, the disk re-allocates each block back into the BAM. Collect is complete once all directory entries (PRG, SEQ, REL and USR) have been examined.

Improperly closed files are thrown away by the Collect operation. An improperly closed file is indicated by an asterisk (*) preceding the file type in a directory listing. This can occur in any number of ways; no DCLOSE or CLOSE command after recording a file; DISK FULL occurring before the file is CLOSEd; hitting STOP while saving a program; or a power failure while storing data.

Regardless of how it happens, unclosed files should NOT be SCRATCHED! As you know, SCRATCH does not erase blocks, it merely de-allocates them from the BAM. This means that the old data is left behind (including track & sector chain pointers) but in blocks that are now available for re-use.

Consider this: You pull out a full or almost full diskette. The diskette has no improperly closed files. Now you want to save a couple of programs on this diskette but there's not enough room. So you SCRATCH 4 or 5 old files that are no longer needed. With more than enough space you SAVE your first new program . . . no problem. Now you go to save the second program and for some reason the operation is aborted (DISK FULL, STOP key, etc.), leaving this file improperly closed! Chances are that the last block to be written points at a block that was previously used by one of your old files. This block would contain old track & sector pointers which might point at other blocks that are now in use by (quite possibly) the program that you just saved successfully. SCRATCHing this unclosed file would then de-allocate blocks that were just written PLUS blocks, belonging to your other program. Another SAVE at this point could be hazardous. The disk might choose to re-use those free blocks that belong to the other program, thus replacing parts of the first program with parts of the second.

A COLLECT after the aborted SAVE would have avoided all problems. The unclosed PRG file would be discarded, and the integrity of the other files preserved. Some believe that reported problems with write & replace (using the '@' symbol) are connected somehow to the presence of unclosed files, but no proof is available.

Collect has only one drawback. Any blocks allocated by the block-allocate (B-A) command will be freed by Collect as these will not belong to a chain as with other files. Subsequent B-A & B-W commands will use these blocks, possibly overwriting valid data. However, with the advent of Relative files, direct access should be fading from use.

Otherwise, it's never too soon for a Collect. If your block count doesn't add up or you suspect another undesirable condition, use Collect to be safe. ■

RTQL

A SELF-COMPILING RELATIONAL DATABASE

DATABASE SELECTION—

- Uses screen masks to form query
- Provides extensive search capabilities
- Search arguments can include arithmetic/boolean functions, multi-field comparisons
- Queries can generate input for automatic database maintenance
- Queries can be stored in "Query Library" and executed from menu on demand
- Any number of fields can be queried concurrently
- Query output can be routed to disk, CRT report formatter, VISICALC™ or SOURCE™

DATABASE MAINTENANCE—

- Uses sophisticated screen formatting & data entry, like on IBM 3270's!
- Generates it's own screens automatically!
- Handles records up to 4K in length, using multiple screen "Pages"!
- Automatic data compression for increased disk capacity
- Uses SuperKram access method for incredibly fast access, LESS THAN .2 SECONDS FOR A RECORD!
- Automatic index creation/maintenance
- Automatic maintenance capabilities
- "Goof-Proof" error handling
- Input can come from VISICALC™ or SOURCE™

DATABASE REPORTING—

- Automatic headlines
- Automatic field editing
- Report fields can be calculated, sub-totaled & cross-footed in any manner desired
- Optional counter breaks may be set
- Automatic grand totals
- Automatic statistics

REQUIREMENTS —

SuperKram (available separately) and: Commodore Pet 32K (40 or 80 col.) and 2040/4040/8050 disk.

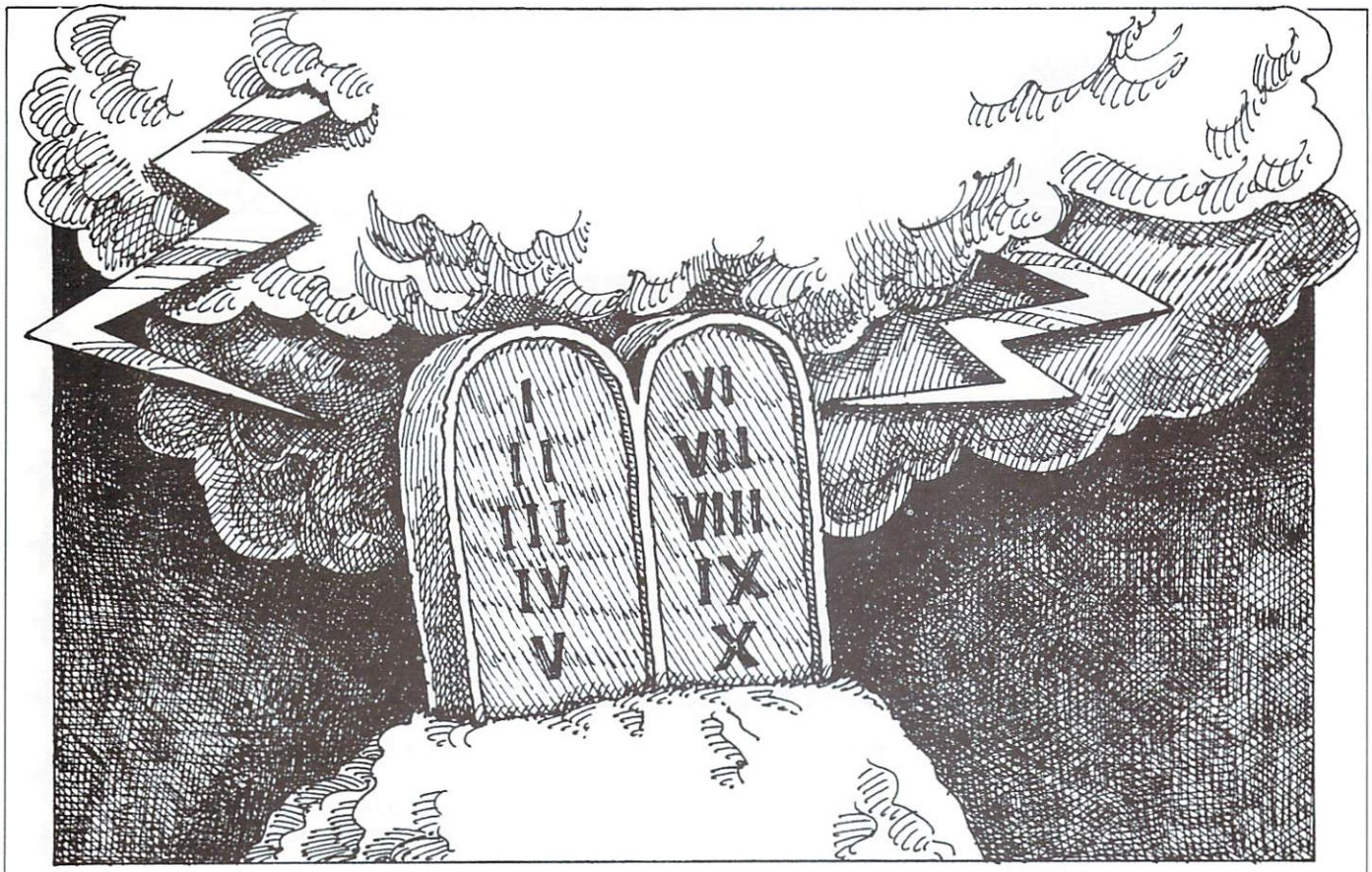
ONLY $225

USA UNITED SOFTWARE OF AMERICA

750 THIRD AVE.
NEW YORK NY 10017

Look for the RED-WHITE-BLUE United Software Display at your local computer dealer, or send check or moneyorder, plus $3.00 shipping

(212) 682-0347 Telex 640055



STANDARDS!!

In the last few years, I have come to recognize a problem which seems to be reaching near critical proportions in our industry. STANDARDS, or actually, a lack of them. You've all heard people talk about this, because I know I'm not the first. Probably you all have run into a problem which, somehow or another, could ultimately have been attributed to standards.

You know that program you use every day which always seems to require a different answer to get back to the menu screen. Or how about that manual you just bought that is totally useless because you can't find anything it's purported to contain. Yes, these are standards problems, and we've been putting up with them because after awhile you learn the "ins and outs" of finding your answers in that book and you've finally learned all 27 responses to get back to the menu.

Why does this problem exist? Some people speculate that because of the extreme growth in the computer industry, highly technical people, with no communications background, became a large percentage of the workforce. Other people stress that because of the technological advances made every year, there is no sense setting standards when they will become quickly outdated. Okay, so we're not the greatest writers. And it does seem hardly worth it to sit down and create a beautiful application when its market life is less than a year. So maybe we shouldn't worry about standards, right? Wrong!!

What is a standard anyway? A "standard" can be described as something established by authority, custom, or general consent as a model or example. Why are standards developed? Because they give people something to follow, without having to make a decision every time the same situation is encountered. Imagine if Daniel Webster hadn't set a standard for the publication of his dictionary. It wouldn't have sold very well if the words weren't in alphabetical order, would it? A simple standard, but yet without it, a dictionary is just about useless.

Setting standards for documentation and programming doesn't seem to be that hard to implement, yet the number of packages, manuals, and books that contain standards is very small. Actually there isn't a total lack of standards. There are some companies who really believe in them and they are probably partly successful because of it. And if you take a look at the best selling packages on the PET/CBM, you'll immediately notice that standards were set up in the beginning, and carried through in every facet and module of the package. You'll also notice that the accompanying documentation is also easy to follow.

So do we all have to go back to school and take some creative writing courses? No, in fact that really wouldn't solve the problem at all. What would happen would be that we would see a flourishing of "user friendly" manuals and programs that nobody could follow. No you don't have to be a good writer or even a real good programmer to put together a good, easy-to-use package. What it takes is a set of standards that are followed from documentation through to programming.

How can this be done? There are probably many possibilities but there is one concept that has been developed to assist in the implementation of Information Processing Systems. This concept, called the "Life Cycle Process," was initially developed for the implementation of manual systems, but has been brought forward and used by industry to implement computerized information processing systems. The reason I even discuss it here is because the single most important attribute of this process is the development of standards. And this happens a long time before the programming even begins.

The Life Cycle has basically four components: The Study Phase, The Design Phase, The Development Phase, and finally The Operations Phase. It should be noted that the first two phases comprise about 70% of the total time spent working on the project. What this means is that programming and installation make up a small percentage in relation to the total project time. Sounds crazy, but the companies using this technique claim that if a problem is studied, and a solution worked out on paper (via flowcharts, etc.), the amount of time spent programming and maintaining the system after the fact is negligible. This latter fact, maintenance, should strike home to many people.

Another fact that I found hard to believe when reading about this methodology is that by the time the programming was started, the documentation was already written. The only thing necessary at installation time was to make refinements here and there.

So where do standards fit into all of this anyway? Well let's start with documentation. By the time you start to write your documentation, you know what problem your system is going to solve. Documentation actually starts at the stage when you are developing a method to solve this problem. Before you start this process, decide on a format for the documentation and set some standards. Do you need a "Table of Contents?" How about an "Index?" Does the documentation need to be in "Chapter" format?

When writing documentation, remember the intended "end user." Will the "end user" even read the documentation or will they regard it as a reference to "run to" when there is a problem. Studies in this area show that if this is the case, the documentation should be in the form of reference material, that is, answers easy to find with as little reading as possible. Then again, the "end user" may be one who would read the manual and never refer to it again when operating the system. In this case, then, a very "user friendly" document is appropriate.

I've often heard manufacturers, authors, software people, etc. explain that if the user would have "read" the documentation, then the problem wouldn't have happened. I say that if the documentation or program was written "for" the "intended end user," the problem may have never occurred. In any case, the problem is never the user's—it's your's!

Now that you have set some standards for the manual, you can start documenting. What is the first step in solving your problem? Aha! Chapter 1. Now what? Next chapter. And so on.

As you are going through this phase, you will be setting standards for the actual programs themselves. Should the user always press <RETURN> after entering data? If so, stick to that standard throughout, and make sure the programmer writing the program has implemented this. Should you always display "IS EVERYTHING CORRECT (Y/N)" when you want to prompt for changes, or is "CHANGES (Y/N)" more appropriate? Whatever the decision, stick to it throughout the package.

Remember a program with no standards is as hard to follow as a dictionary that is not in alphabetical order.

Once your documentation is finished, you now have a "standards manual" to go by when writing the program. The written documentation is actually a design document upon which the programming should be based.

You may not agree with the idea of writing documentation beforehand. In fact you always set standards and have a design phase before programming. Then you write the documentation and it still comes out good. The only answer I have to that is that you probably would save a lot of time if you documented as you went through the design phase.

Programming standards. I know you have all been grilled about this one. And probably for good reason—a product turns out looking good or bad because of the programming. Again you have to set standards and follow ones already set by the system design. If you don't, some user will be cursing you out at the first inconsistency!

Then there are the popular standards of writing programs. One of the most popular methods touted today is the "structured programming method." Of course you have to implement it the most efficient way for the hardware you are working with. But don't write it off; it has its paybacks time and again. And the hardware and operating systems are approaching a point where using this technique is just as efficient as not using some other optimizing approach. And maintenance programming is definitely easier when using this method.

Comments! This should be the number one programming standard. Without them, it is sometimes hard for even the original programmer to find a bug. And don't do it after the fact; do it as you go along when the flow of your program is fresh in your mind.

As a final note: make standards the most important thing when writing documentation, designing a system, writing a book, or writing a program. If you do, you will realize the benefits immediately. ■

—Dave Scott

Accessing the SuperPET RS-232 Port

Using the SuperPET's RS-232 serial port from the 6809 processor is made easy because of communication routines included in ROM and the SETUP menu which allows selection of stop-bit, parity and baud rate options. However, one of the great mysteries of the SuperPET has been how to access the RS-232 serial port using the 6502 processor. So, this month we will try to remove some of that mystery by providing details of register addresses and formats for programming the serial port.

The serial port on the SuperPET is a 6551 Asynchronous Communication Interface Adapter (ACIA) manufactured by Commodore Semiconductor Group. The ACIA uses a single +5 volt power supply, features an on-chip baud-rate generator and is capable of half-duplex or full-duplex operation. Word length, number of stop-bits, parity generation/checking and baud-rate are all programmable.

The ACIA is seen by both SuperPET microprocessors as four memory locations at address (hexadecimal) $EFF0-$EFF3. Input/output and programming of the ACIA is done by writing to or reading from these addresses as shown in Table 1.

Table 1. SuperPET ACIA Memory Locations

Memory Address	WRITE Access	READ Access
$EFF0	Fill Transmitter Data Register	Unload Receiver Data Register
$EFF1	Programmed Reset (Use any data)	Read Status Register
$EFF2	Program Command Register	Read Command Register
$EFF3	Program Control Register	Read Control Register

Table 2. ACIA Control Register Programming

The Control Register is used to select the desired operating mode for the ACIA. The word-length, number of stop-bits, clock control and baud-rate are all programmed via the Control Register as shown in Table 2.

Control Bits	Control Function	Valid Data Values
7	Number of Stop Bits	0 = 1 Stop Bit 1 = 2 Stop Bits
6–5	Set Word Length (1.5 stop bits if 5 bits + Parity)	00 = 8 bits 01 = 7 bits 02 = 6 bits 03 = 5 bits
4	Select Clock Source (Always set to "1")	0 = External Clock 1 = Baud-rate Generator
3–0	Select Baud-rate	$0 (Hex) = Not Used $1 = 50 Baud $2 = 75 $3 = 110 $4 = 134.5 $5 = 150 $6 = 300 $7 = 600 $8 = 1200 $9 = 1800 $A = 2400 $B = 3600 $C = 4800 $D = 7200 $E = 9600 $F = 19200

Table 3. ACIA Command Register Programming

The Command Register in the 6551 ACIA is used to control parity generation/checking, receiver echo and transmit/receive functions as shown in Table 3.

Command Bits	Command Function	Valid Data Values
7–5	Set Parity Options	xx0 = Parity Disabled 001 = Odd Parity on Xmit & Recv 011 = Even Parity on Xmit & Recv 101 = Mark Parity Xmit Recv Parity Disabled 111 = Space Parity Xmit Recv Parity Disabled
4	Set Normal/Echo Mode	0 = Normal (No Echo) 1 = Echo for Receiver
3–2	Transmitter Control	00 = Xmitter Disabled, No Request-to-Send 01 = Xmitter Enabled, Request-to-Send 10 = Xmitter Disabled, Request-to-Send 11 = Xmitter Disabled, Request-to-Send (Transmit BRK)
1	Receiver Interrupt Enable	0 = Interrupt Enabled from Status Register Bit 0 1 = Interrupt Disabled
0	Data Terminal Ready	0 = Disable Recvr/Xmitter 1 = Enable Recvr/Xmitter

Table 4. ACIA Status Register Definitions

The Status Register is a read-only register which provides the processor with the status of various ACIA functions. The format of the Status Register is outlined in Table 4.

Status Bits	Status Functions and Values	
0*	1 = Parity Error Detected	0 = No Parity Error
1*	1 = Framing Error Detected	0 = No Framing Error
2*	1 = Overrun Has Occurred	0 = No Overrun
3	1 = Receiver Data Register is Full	0 = Receiver Not Full
4	1 = Transmitter Data Register is Empty	0 = Xmitter Not Empty
5	1 = No Data Carrier	0 = Carrier Detected
6	1 = Data Set Not Ready	0 = Data Set Ready
7	1 = Interrupt Requested	0 = No Interrupt Request
*	No Interrupt Request occurs for these status conditions ■	

— Dave Middleton

Weekday Calculator

This neat little subroutine returns the day of the week for any date given in DAY/MONTH/YEAR format. Of course you could change it around for YEAR/MONTH/DAY—just alter the order of the variables following the INPUT statement. The program does not check for date validity . . . but that's no problem. Just do some testing for day greater than 31 some months, 30 other months and 28 for February. For leap years, do an extra test of YEAR/4 = INT(YEAR/4) in the case of Feb. 29.

```
100 INPUT "DD, MM, YYYY";D,M,Y
110 K = INT( (60 + (100/M) )/100 )
120 F = 365 * Y + D + 31 * (M - 1) -
  INT(.4*M + 2.3) * (1 - K)
130 F = F + INT((Y - K/4) -
  INT(.75*( INT((Y - K) / 100 + 1)))
140 F = F - INT(F/7) * 7
150 PRINT MID$
  ( "SATSUNMONTUEWEDTHUFRI",
  F * 3 + 1, 3 ) ■
```

Keyed Random Access for the PET/CBM

by
Glen Pearce
Commodore Johannesburg

Since the advent of Relative Files and the large storage capacity of the CBM 8050 Disk, some form of 'K.R.A.' (Keyed Random Access) would be useful to make full use of these facilities. Here is a version that meets most of the specifications of K.R.A., but is relatively (excuse the pun!) easy to use. It works as follows:

An ordinary sequential file is used to store a 'key-file' of all records held within a system (e.g. Stock, Accounts, Clients, etc.). This key-file would normally contain the first 10 characters of a customer's name (Part #, Account #, etc.) followed by *the Relative Record Number* of the record containing the remaining data for that customer.

Now, all you have to do is search through this key-file until you find the record you're looking for; retrieve the relative record number and you have access to the main record. The only problem in doing this in BASIC is time—especially if you have 500 to 1000 records or more!

Here is a machine-code routine which will do the above significantly faster (it searches through 500 ten-character record keys in approximately 4 seconds). This routine may only be used with BASIC 4.0 and DOS 2.0. Here's how you use it:

The length of each record in the key-file (SEQ) is not important and it may contain any valid ASCII characters (for safety's sake, stick to alpha-numerics only). To separate the record-key from the associated relative record number, a delimiter must be used. In this case the delimiter is a '#' symbol. Therefore, a record in the SEQ key-file should look something like:

SMITH# 1234

The space between the delimiter and the rel/rec number is the sign of the number and can be suppressed if space-saving on the disk is necessary.

It is important that each record in the key-file be separated by a Carriage Return—CHR$(13). This shouldn't present any problem as the PET/CBM automatically sends this character after each PRINT# command.

The K.R.A. machine code program must be located at the top of memory and protected in the usual way:

POKE 53, 127:POKE 52, 0:CLR

. . . must be the first statement in your program.

This program also allows you to do a form of "pattern-matching." Say, for instance, you don't know the exact spelling of a record-key in the key-file. All you do is enter the first few characters of the record-key and allow the program to search for that. When a 'match' is found in the file, the attached rel/rec number will be returned. You could then retrieve that relative record and display it. If it is NOT the correct record, simply tell the program to continue searching the key-file until it finds another match and so on. If NO match is found, a relative record number of 0 (zero) will be returned by the K.R.A. routine.

Here is an example of a BASIC program using the routine:

```
100 A$ = "":A = 0:REM INITIALIZE VARIABLES
    BEFORE USING K.R.A.
110 INPUT "ENTER SEARCH-STRING";A$
120 DOPEN#2, "KEY-FILE":IF DS< >0 THEN
    PRINT DS$:STOP
130 SYS 32512, 2, A$, A
140 IF A = 0 THEN DCLOSE#2:STOP:REM NO
    MATCH
150 REM RETRIEVE THE ASSOCIATED RELATIVE
    RECORD
160 REM AT THIS STAGE, IF THE REL/REC IS NOT
    CORRECT
170 REM YOU COULD 'GOTO 130' TO LOOK FOR
    ANOTHER MATCH
```

Any string and numeric variable may be used, but should be declared before the SYS 32512 to the routine. (In the above example 'A$' would have been initialized by the INPUT statement.) The '2' used after the first comma in the SYS command is the logical file number used in the DOPEN statement. It is important to check the DISK STATUS word (DS) after opening the file.

Adding records to the key-file could be a problem once the file gets large. Make use of the APPEND# command in BASIC 4.0 to simply append new record-keys to the file.

Another suggestion is to have separate key-files. For alphabetic keys there would be 26 titled 'A' to 'Z'; for numeric keys, 10 labelled '0' to '9'; or combine for alpha-numeric and have 36 separate key files. Now you could simply check the first character of the search string (i.e., LEFT$(A$,1)) and open that particular file. This would reduce your key-file size to approximately 100 records per file in a 2000 record system, thereby making your search times even faster! ➡

```
30      REM   ********************************************************************************
40      REM   *                                                                              *
50      REM   *      BASIC LOADER FOR MACHINE CODE ISAM ROUTINE                              *
60      REM   *                    GLEN PEARCE 20/8/81                                       *
70      REM   *                                                                              *
80      REM   ********************************************************************************
90      REM
100     POKE53,127:CLR:REM LOWER MEMTOP TO PROTECT PROGRAM
110     FOR I=32512 TO 32767:READ J:POKE I, J:NEXT:END
200     DATA      32,    73,   127,    32,    45,   201,   165,    18,   240,     3
210     DATA      76,     0,   191,   165,    17,   133,   210,    32,    82,   127
220     DATA     166,   210,    32,   198,   255,   160,     0,    32,   228,   255
230     DATA     166,   150,   208,    66,   201,    13,   240,   243,   209,     1
240     DATA     208,    18,   200,   196,     0,   144,   236,    32,   228,   255
250     DATA     166,   150,   208,    46,   201,    35,   240,    90,   208,   243
260     DATA      32,   228,   255,   166,   150,   208,    33,   201,    13,   240
270     DATA     210,   208,   243,    32,   245,   190,    32,   152,   189,   160
280     DATA       0,    96,    32,    73,   127,   177,    68,   133,     0,   200
290     DATA     177,    68,   133,     1,   200,   177,    68,   133,     2,    96
300     DATA      32,    73,   127,   169,     0,   133,    95,   133,    96   133
310     DATA       7,   162,   144,    32,   122,   205,   160,     0,   165,    94
320     DATA     145,    68,   200,   165,    95,    41,   127,   145,    68,   200
330     DATA     165,    96,   145,    68,   200,   165,    97,   145,    68,   200
340     DATA     165,    98,   145,    68,    32,   204,   255,    96,    32,    73
350     DATA     127,   169,     0,   133,    95,   133,     7,    32,   195,   127
360     DATA     201,    13,   240,    23,   166,   150,   208,   188,   133,    96
370     DATA      32,   195,   127,   201,    13,   240,    10,   166,   150,   208
380     DATA     175,    32,   213,   127,    76,   170,   127,   162,   144,    32
390     DATA     122,   205,    76,   116,   127,    32,   228,   255   201,    13
400     DATA     240,    10,   201,    48,   144,   245,   201,    58,   176,   241
410     DATA      41,    15,    96,   133,     0,   165,    95,    72,   165,    96
420     DATA      72,     6,    96,    38,    95,     6,    96,    38,    95,   104
430     DATA     101,    96,   133,    96,   104,   101,    95,   133,    95,     6
440     DATA      96,    38,    95,   165,     0,   101,    96,   133,    96,   169
450     DATA       0,   101,    95,   133,    95,    96
```

```
LINE# LOC  CODE LINE

0001  0000             ;************************************************
0002  0000             ;*  SEARCH THRU A SEQ FILE FOR A KEY RECORD AND  *
0003  0000             ;*   THEN RETRIEVE AN ATTACHED REL/REC NUMBER.   *
0004  0000             ;*                                               *
0005  0000             ;*    GLEN PEARCE                    22/08/81    *
0006  0000             ;*    COMMODORE, JOHANNESBURG, SOUTH AFRICA      *
0007  0000             ;************************************************
0008  0000             ;
0009  0000             ; ## CONSTANTS FROM PET BASIC (BASIC 4.0) ##
0010  0000             GETCHR = $FFE4          ;GET A CHARACTER
0011  0000             CLRCHN = $FFCC          ;CLOSE I/O CHANNELS
0012  0000             COIN   = $FFC6          ;SET INPUT DEVICE
0013  0000             CHKCOM = $BEF5          ;CHK FOR COMMA
0014  0000             FRMEVL = $BD98          ;EVALUATE EXPRESSION
0015  0000             FACINT = $C92D          ;CONVERT FL/P TO INT
0016  0000             SNERR  = $BF00          ;PRINT SYNTAX ERROR
0017  0000             ;
0018  0000             ; ## PAGE ZERO VARIABLES ##
```

```
0019  0000                  LENGTH = $00          ;TEMP STORE OF STR LENGTH
0020  0000                  WORK1  = $01          ;TEMP WORK AREA
0021  0000                  CHKINT = $11          ;CHECK FOR INTEGER
0022  0000                  CURFIL = $D2          ;CURRENT FILE NUMBER
0023  0000                  VARPNT = $44          ; PNTR TO CURRENT VARIABLE
0024  0000                  FAC    = $5E          ;MAIN FLT/PNT ACCUMULATOR
0025  0000                  ;
0026  0000                         * = $7F00
0027  7F00                  ;
0028  7F00  20 49 7F       FIND   JSR EVALEX     ;CHK SYNTAX OF COMMAND
0029  7F03  20 2D C9              JSR FACINT     ;IN BASIC LINE & EXTRACT LFN
0030  7F06  A5 12                LDA CHKINT+1   ;AND SEARCH STRING
0031  7F08  F0 03                BEQ ISINTG
0032  7F0A  4C 00 BF              JMP SNERR      ;EXIT IF SYNTAX ERROR
0033  7F0D  A5 11       ISINTG LDA CHKINT
0034  7F0F  85 D2                STA CURFIL     ;SET UP LFN FOR READ
0035  7F11  20 52 7F              JSR FNDEXP     ;FIND SRCH STRING
0036  7F14  A6 D2                LDX CURFIL
0037  7F16  20 C6 FF              JSR COIN       ;SET I/O FOR READ
0038  7F19                  ;
0039  7F19  A0 00          GET10  LDY #0
0040  7F1B  20 E4 FF       GET11  JSR GETCHR     ;GET CHAR FROM FILE
0041  7F1E  A6 96                LDX $96        ;CHK STATUS BYTE FOR EOF
0042  7F20  D0 42                BNE DONE1
0043  7F22  C9 0D                CMP #13        ;CHK FOR C/RET
0044  7F24  F0 F3                BEQ GET10      ;MOVE TO NEXT RECORD
0045  7F26  D1 01                CMP (WORK1)Y   ;COMPARE TO EQUIVALENT
0046  7F28  D0 12                BNE CLRSTR     ;CHAR OF SEARCH STRING
0047  7F2A  C8                   INY
0048  7F2B  C4 00                CPY LENGTH     ;IF NUMBR OF CHARS CHK'D
0049  7F2D  90 EC                BCC GET11      ;EQUALS LEN OF SEARCH STRING
0050  7F2F  20 E4 FF       FNDDEL JSR GETCHR     ;THEN MATCH IS MADE
0051  7F32  A6 96                LDX $96
0052  7F34  D0 2E                BNE DONE1
0053  7F36  C9 23                CMP #'#        ;FIND DELIMITER & THEN GO
0054  7F38  F0 5A                BEQ RELNUM     ;AND READ IN REL/NO.
0055  7F3A  D0 F3                BNE FNDDEL
0056  7F3C  20 E4 FF       CLRSTR JSR GETCHR     ;DISCARD REST OF STRING
0057  7F3F  A6 96                LDX $96
0058  7F41  D0 21                BNE DONE1
0059  7F43  C9 0D                CMP #13
0060  7F45  F0 D2                BEQ GET10      ;GO AND CHK NEXT STRING
0061  7F47  D0 F3                BNE CLRSTR
0062  7F49                  ;
0063  7F49  20 F5 BE       EVALEX JSR CHKCOM     ;CHK FOR COMMA
0064  7F4C  20 98 BD              JSR FRMEVL     ;& EVALUATE EXPRESSION
0065  7F4F  A0 00                LDY #0
0066  7F51  60                   RTS
0067  7F52                  ;
0068  7F52  20 49 7F       FNDEXP JSR EVALEX     ;FIND SRCH STRING
0069  7F55  B1 44                LDA (VARPNT)Y  ;SET UP STRING PNTRS
0070  7F57  85 00                STA LENGTH     ;IN TEMP WORK AREAS
0071  7F59  C8                   INY
0072  7F5A  B1 44                LDA (VARPNT)Y
0073  7F5C  85 01                STA WORK1
0074  7F5E  C8                   INY
0075  7F5F  B1 44                LDA (VARPNT)Y
```

```
0076  7F61  85 02          STA WORK1+1
0077  7F63  60             RTS
0078  7F64           ;
0079  7F64  20 49 7F  DONE1 JSR EVALEX      ;IF NO MATCH FOUND THEN
0080  7F67  A9 00          LDA #0           ;RETURN A REL/NO. OF ZERO
0081  7F69  85 5F          STA $5F
0082  7F6B  85 60          STA $60
0083  7F6D  85 07          STA $07          ;SET VARIABLE TYPE TO NUMERIC
0084  7F6F  A2 90          LDX #$90
0085  7F71  20 7A CD       JSR $CD7A        ;CONVERT HEX TO FL/P
0086  7F74  A0 00  DONE2    LDY #0
0087  7F76  A5 5E          LDA FAC          ;TRANSFER BCD VALUE OF
0088  7F78  91 44          STA (VARPNT)Y    ;REL/NO. TO NUMERIC VAR
0089  7F7A  C8             INY              ;SPECIFIED IN SYS CMD
0090  7F7B  A5 5F          LDA FAC+1
0091  7F7D  29 7F          AND #$7F         ;STRIP OFF SIGN
0092  7F7F  91 44          STA (VARPNT)Y
0093  7F81  C8             INY
0094  7F82  A5 60          LDA FAC+2
0095  7F84  91 44          STA (VARPNT)Y
0096  7F86  C8             INY
0097  7F87  A5 61          LDA FAC+3
0098  7F89  91 44          STA (VARPNT)Y
0099  7F8B  C8             INY
0100  7F8C  A5 62          LDA FAC+4
0101  7F8E  91 44          STA (VARPNT)Y
0102  7F90  20 CC FF       JSR CLRCHN       ;CLEAR ALL I/O CHANS AND
0103  7F93  60             RTS              ;EXIT PROGRAM
0104  7F94           ;
0105  7F94  20 49 7F  RELNUM JSR EVALEX     ;FIND VARIABLE FOR REL/NO.
0106  7F97  A9 00          LDA #0
0107  7F99  85 5F          STA $5F
0108  7F9B  85 07          STA $07
0109  7F9D  20 C3 7F       JSR NEWDIG       ;READ IN REL/NO. AND CONVERT
0110  7FA0  C9 0D          CMP #13          ;IT TO A 2-BYTE HEX DIGIT
0111  7FA2  F0 17          BEQ PUTVAR
0112  7FA4  A6 96          LDX $96
0113  7FA6  D0 BC          BNE DONE1
0114  7FA8  85 60          STA $60
0115  7FAA  20 C3 7F  NXTDIG JSR NEWDIG
0116  7FAD  C9 0D          CMP #13
0117  7FAF  F0 0A          BEQ PUTVAR
0118  7FB1  A6 96          LDX $96
0119  7FB3  D0 AF          BNE DONE1
0120  7FB5  20 D5 7F       JSR ASCHEX
0121  7FB8  4C AA 7F       JMP NXTDIG
0122  7FBB  A2 90  PUTVAR  LDX #$90
0123  7FBD  20 7A CD       JSR $CD7A
0124  7FC0  4C 74 7F       JMP DONE2
0125  7FC3           ;
0126  7FC3  20 E4 FF  NEWDIG JSR GETCHR     ;GET NEXT REL/NO. DIGIT
0127  7FC6  C9 0D          CMP #13
0128  7FC8  F0 0A          BEQ ENDDIG
0129  7FCA  C9 30          CMP #$30         ;CHK FOR NUMERIC
0130  7FCC  90 F5          BCC NEWDIG
0131  7FCE  C9 3A          CMP #$3A
0132  7FD0  B0 F1          BCS NEWDIG
0133  7FD2  29 0F          AND #$0F         ;MASK OUT THE FOUR MSB'S
```

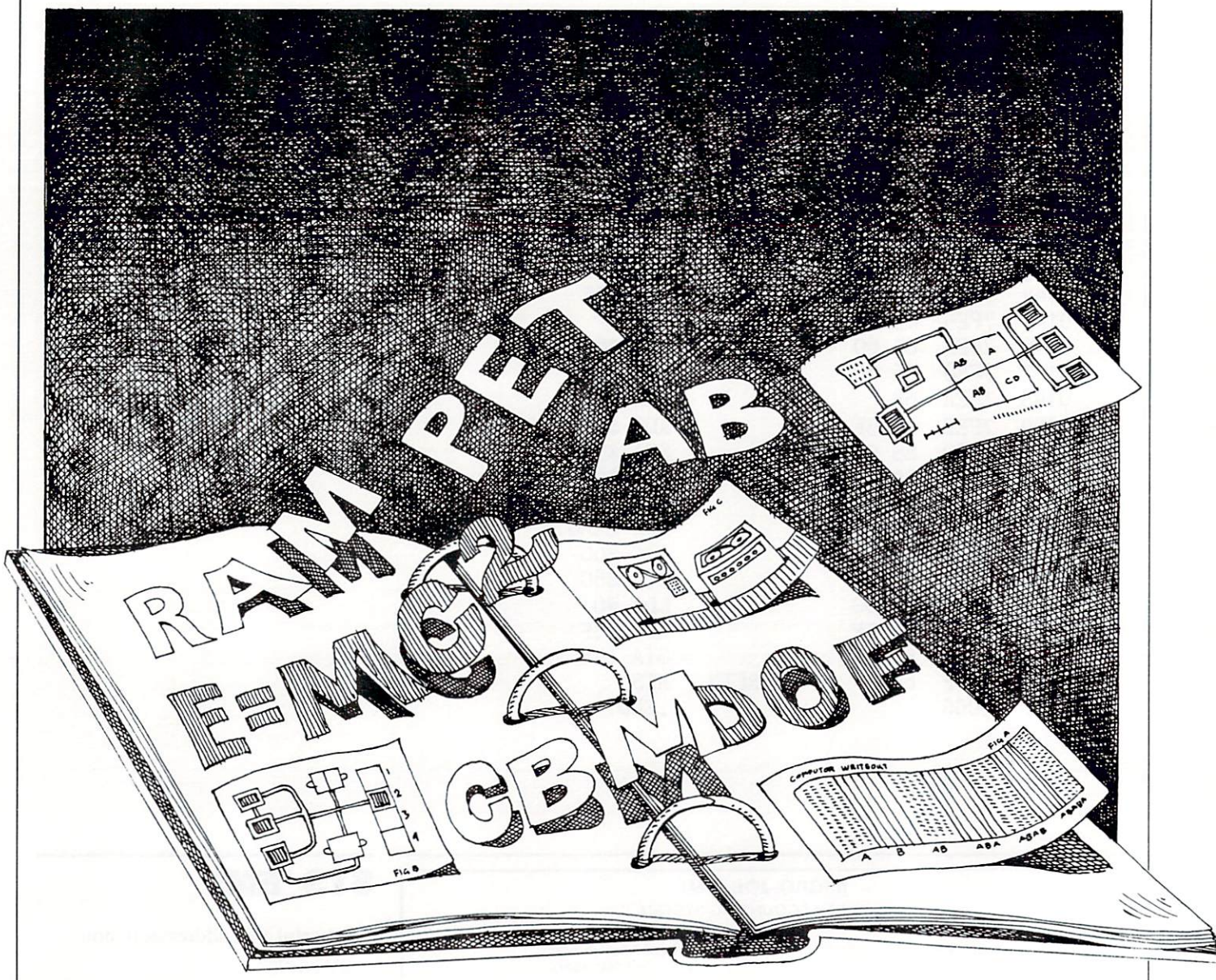
```
0134  7FD4  60          ENDDIG RTS
0135  7FD5              ;
0136  7FD5  85 00       ASCHEX STA LENGTH      ;HANDLE ASC - HEX CONVERSION
0137  7FD7  A5 5F              LDA $5F
0138  7FD9  48                 PHA
0139  7FDA  A5 60              LDA $60
0140  7FDC  48                 PHA
0141  7FDD  06 60              ASL $60
0142  7FDF  26 5F              ROL $5F
0143  7FE1  06 60              ASL $60
0144  7FE3  26 5F              ROL $5F
0145  7FE5  68                 PLA
0146  7FE6  65 60              ADC $60
0147  7FE8  85 60              STA $60
0148  7FEA  68                 PLA
0149  7FEB  65 5F              ADC $5F
0150  7FED  85 5F              STA $5F
0151  7FEF  06 60              ASL $60
0152  7FF1  26 5F              ROL $5F
0153  7FF3  A5 00              LDA LENGTH
0154  7FF5  65 60              ADC $60
0155  7FF7  85 60              STA $60
0156  7FF9  A9 00              LDA #0
0157  7FFB  65 5F              ADC $5F
0158  7FFD  85 5F              STA $5F
0159  7FFF  60          RETN   RTS
0160  8000                    .END


ERRORS = 0000
```


SYS 'EM!

Two useful SYS addresses to note:

SYS 64790
SYS 54386

The first does a jump to 'warm start'—as if turning the machine off and back on again, but without that nasty power interruption. The second can be extremely handy when you want to send an M.L.M. memory dump to the printer. It seems that breaking to the monitor with SYS 4 cancels any CMD status you may have set up previously. ■



Excerpts from a Technical Notebook

DOS 1.2 Problems

1. Save with Replace.
This command sometimes fails but the cause is not really known. It usually only happens on disks which have a lot of files and not a great deal of free space. The effect is to chain into other programs or files on the disk.

2. Rename.
This fails occasionally even though the disk system gives the '00,ok,00,00' message, the cause once again is not really known but it may fail due to there being scratched entries on the disk or the number of disk entries is a multiple of eight (ie. a full block).

3. Duplicate.
If a disk is removed while doing a duplicate there is a very good chance that it will be totally corrupted so make sure that the disks to be duplicated are in the correct drive before starting the command.

4. Write protect tab.
Writing to a disk with a write protect tab is attempted. Then when a command is later given to read the disk, even if it has a write protect tab on, at least one write will be made. The solution is to power the disk drive down completely and then start again.

5. Sequential files.
If a sequential file of 254 characters (or any multiple) is written to the disk then an extra carriage return is added to the end of the file.

6. Block Allocate and Block Free.
The best way to use this command is to convert numbers

into strings and concatennate this to the command before sending the command down the error channel.

7. Illegal Track and Sector.
If illegal track or sector command parameters are given to the block commands then partial overlaying of error messages results.

8. Block free.
If an unallocated block is freed, the block count is automatically incremented by one and thus an incorrect number of blocks free can be generated ie. more than 670! Validate will restore the correct number of blocks.

9. Validate 1.
If an error occurs while validation of a diskette is taking place, then the BAM will be left in an indeterminate state. Re-initialization of the diskette is necessary in order to restore the disk.

10. Validate 2.
The validate command frees any sectors allocated for random access.

11. Save and Open without giving a drive number.
This causes partial updating on both drives, thus corrupting both BAMs. This bug is probably the cause of more disk corruption problems than all the others put together, and may actually be the cause of some failures such as save with replace.

12. DOS handling of the IEEE bus.
Occasionally during multiple 'GET' the disk unit transmits a data byte onto the bus, even when the PET has Attention high. This gives the appearance that the PET is sending a command to all other peripherals on the bus.

13. Using asterisk as the filename.
An asterisk may be used as the filename to access the last LOADed or SAVEd program. If the last program was saved with replace, then the asterisk accesses the old version of the program (which has now been scratched from the directory) rather than the program which was just saved.

14. Memory Read.
The byte returned by a memory read operation is not accompanied by a carriage return so use GET to access the character.

Relative Records Bug

There is a serious bug in the relative record system on both DOS 2.1 and 2.5. The bug only occurs when two files have been opened for reading and writing. The bug only shows at certain length records and at set distances through the file. The following example demonstrates the bug:-

```
READY
30 DOPEN #1, "KEYTEST", L13, DO
40 FOR J=11 TO 50 - 100
50   A$=STR$(J)+"++++++++++++++++++":A$---MID$
   (A$, 2, 13)
70 RECORD#1, (J)
80 PRINT#1, A$
90 NEXT
100 DCLOSE#1
110 DOPEN#2, "FILETEST", L254, DO
120 FOR J=1 TO 50
130 B$=STR$(J)+"++++++++++++++++++++++++++++++
    "
140 RECORD#2, (J): PRINT #2,B$:NEXT
150 DCLOSE#2
190 DOPEN#1,"KEYTEST",L13,DO
200 FOR J=1 TO 50:INPUT#1,A$:PRINTA$:NEXT
210 DCLOSE#1
220 DOPEN£2,"FILETEST",L254,DO
230 FOR J=1 TO 50: INPUT#2,A$:PRINTA$:NEXT
240 DCLOSE#2
250 PRINT"PRESS A KEY
260 GETZ$:IFZ$=""GOTO260
280 DOPEN#1,"KEYTEST",L13,DO
290 DOPEN#2,"FILETEST",L254,DO
300 X=34
310 FOR J=1 TO X: INPUT#1,A$
320 PRINTA$: NEXT
330 RECORD#2,25
340 INPUT#2,B$
350 PRINTB$
370 FOR J=X TO 50
380 A$=LEFT$(A$,9)+"TEST"
390 RECORD#1,(J)
400 PRINT#1,A$
410 PRINTA$
420 INPUT#1,A$
430 NEXT
440 DCLOSE#1
450 DCLOSE#2
510 DOPEN#1,"KEYTEST",L13,DO
520 FOR J=1 TO 50: INPUT#1,A$:PRINTA$: NEXT
530 DCLOSE#1
```

The program sets up 2 files (30 - 250) with unique records. The first 34 records are read from 'Keytest' then a record is read from 'Filetest'. Now records on 'Keytest' are updated. Both files are then closed (280 - 450). When 'Keytest' is read again some of the updated records are unchanged. In this example, records 34 - 40 are the same as they were originally.

Thus it is not possible to have two relative files open for reading/writing at the same time with any degree of certainty that records will be updated correctly.

There are three solutions to this:-

1. Open and close each file before accessing another.

2. Thoroughly test the record length chosen to see that it does not cause the bug.

3. This solution has no reason for working but it cured the bug in the example program so try it at your own risk: When the files are opened in lines 280 and 290, position the record pointer at record number 1, read it into the PET, reset the record pointer to 1 and then write it out again. The file then reads and updates correctly. Do this for both files.

BASIC 4.0 String Bug

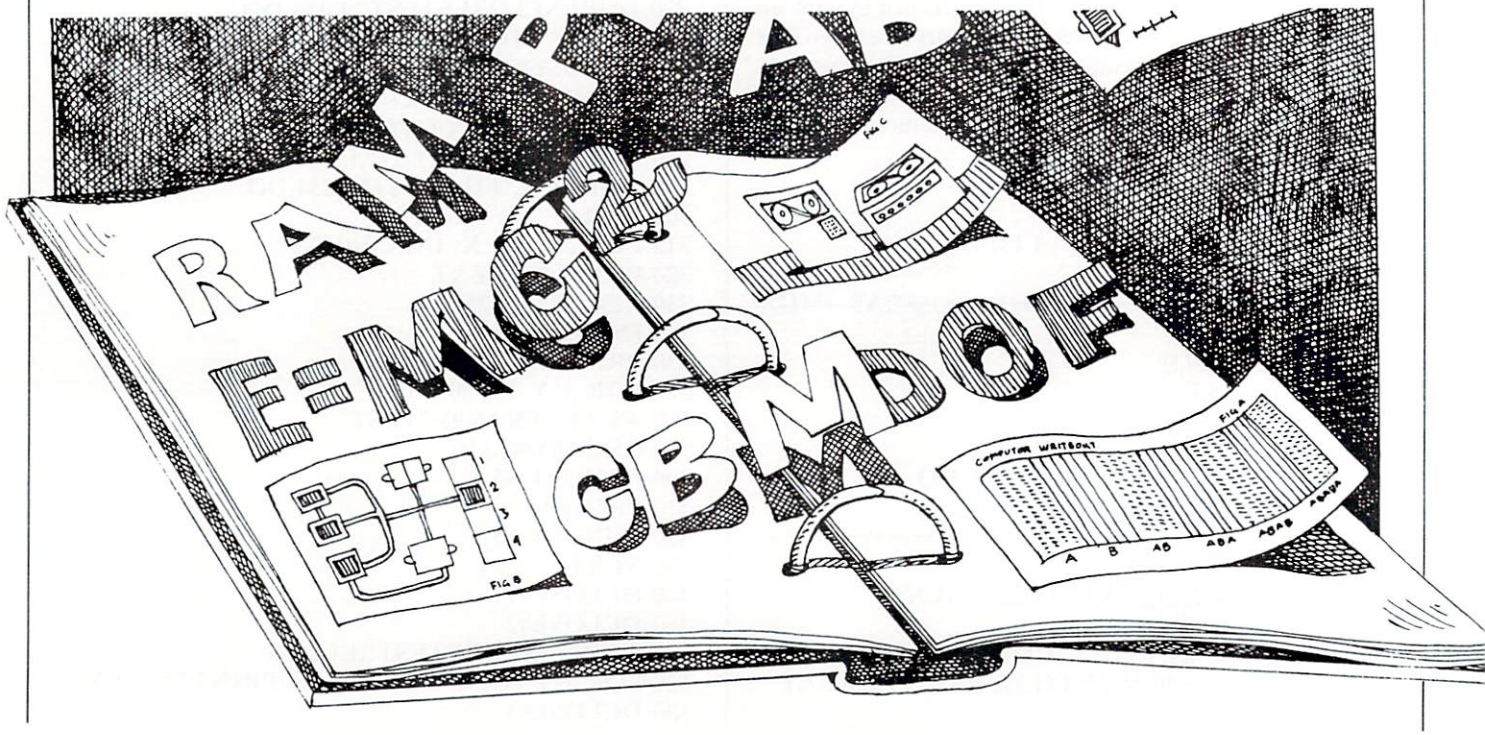
This error only occurs under BASIC 4.0 when there are less than 768 bytes free (or 3 times the largest string size), after all variables and arrays have been assigned by a program. The error is that BASIC fails to detect an 'Out of Memory' condition soon enough, causing corruption of string data and sometimes program text. An example of this bug on a 32k system follows:

```
10 DIM A(6330)
20 BUG$ = BUG$ + "W" + "x" :PRINT BUG$ :GOTO20
```

This program will build a string of alternating characters "WxWxWxWx". It will terminate correctly with an 'Out of Memory in 20' error, but the string will be corrupted after only a few passes.

The easiest solution to the problem is to trap the error from BASIC before it occurs:

```
IF FRE(0) < 768 THEN PRINT "Out of Memory'
:STOP
```



VIC Memory Map

The following memory maps provide a guide which shows which special locations are set aside for use by the VIC's operating system . . . and what those locations are used for.

Memory Map

HEX	DECIMAL	DESCRIPTION
0000	0	Jump for USR
0001-0002	1-2	Vector for USR
0003-0004	3-4	Float-Fixed vector
0005-0006	5-6	Fixed-Float vector
0007	7	Search character
0008	8	Scan-quotes flag
0009	9	TAB column save
000A	10	0 = LOAD, 1 = VERIFY
000B	11	Input buffer pointer/# subscript
000C	12	Default DIM flag
000D	13	Type: FF = string, 00 = numeric
000E	14	Type: 80 = integer, 00 = floating point
000F	15	DATA scan/LIST quote/memory flag
0010	16	Subscript/FNx flag
0011	17	0 = INPUT;$40 = GET;$98 = READ
0012	18	ATN sign/Comparison eval flag
0013	19	Current I/O prompt flag
*0014-0015	20-21	Integer value
0016	22	Pointer: temporary string stack
0017-0018	23-24	Last temp string vector
0019-0021	25-33	Stack for temporary strings
0022-0025	34-37	Utility pointer area
0026-002A	38-42	Product area for multiplication
*002B-002C	43-44	Pointer: Start of Basic
*002D-002E	45-46	Pointer: Start of Variables
*002F-0030	47-48	Pointer: Start of Arrays
*0031-0032	49-50	Pointer: End of Arrays
*0033-0034	51-52	Pointer: String storage (moving down)
0035-0036	53-54	Utility string pointer
*0037-0038	55-56	Pointer: Limit of memory
0039-003A	57-58	Current Basic line number
003B-003C	59-60	Previous Basic line number
003D-003E	61-62	Pointer: Basic statement for CONT
003F-0040	63-64	Current DATA line number
0041-0042	65-66	Current DATA address
*0043-0044	67-68	Input vector

* *Useful memory location*

HEX	DECIMAL	DESCRIPTION
0045-0045	69-70	Current variable name
0047-0048	71-72	Current variable address
0049-004A	73-74	Variable pointer for FOR/NEXT
004B-004C	75-76	Y-save; op-save; Basic pointer save
004D	77	Comparison symbol accumulator
004E-0053	78-83	Misc work area, pointers, etc
0054-0056	84-86	Jump vector for functions
0057-0060	87-96	Misc numeric work area
*0061	97	Accum#1: Exponent
*0062-0065	98-101	Accum#1: Mantissa
*0066	102	Accum#1: Sign
0067	103	Series evaluation constant pointer
0068	104	Accum#1 hi-order (overflow)
*0069-006E	105-110	Accum#2: Exponent, etc.
006F	111	Sign comparison, Acc#1 vs #2
0070	112	Accum#1 lo-order (rounding)
0071-0072	113-114	Cassette buffer length/Series pointer
*0073-008A	115-138	CHRGET subroutine (get BASIC char)
007A-007B	122-123	Basic pointer (within subroutine)
008B-008F	139-143	RND seed value
*0090	144	Status word ST
0091	145	Keyswitch PIA: STOP and RVS flags
0092	146	Timing constant for tape
0093	147	Load = 0, Verify = 1
0094	148	Serial output: deferred char flag
0095	149	Serial deferred character
0096	150	Tape EOT received
0097	151	Register save
*0098	152	How many open files
*0099	153	Input device (normally 0)
*009A	154	Output (CMD) device, normally 3
009B	155	Tape character parity
009C	156	Byte-received flag
009D	157	Direct = $80/RUN = 0 output control
009E	158	Tape Pass 1 error log/char buffer
009F	159	Tape Pass 2 error log corrected
*00A0-00A2	160-162	Jiffy Clock (HML)
00A3	163	Serial bit count/EOI flag
00A4	164	Cycle count
00A5	165	Countdown, tape write/bit count
00A6	166	Pointer: tape buffer
00A7	167	Tape Write ldr count/Read pass/inbit
00A8	168	Tape Write new byte/Read error/inbit cnt
00A9	169	Write start bit/Read bit err/stbit

* *Useful memory location*

HEX	DECIMAL	DESCRIPTION
00AA	170	Tape Scan;Cnt;Ld;End/byte assy
00AB	171	Write lead length/Rd checksum/parity
00AC-00AD	172-173	Pointer: tape buffer, scrolling
00AE-00AF	174-175	Tape end addresses/End of program
00B0-00B1	176-177	Tape timing constants
*00B2-00B3	178-179	Pointer: start of tape buffer
00B4	180	Tape timer (1 = enable); bit cnt
00B5	181	Tape EOT/RS-232 next bit to send
00B6	182	Read character error/outbyte buffer
*00B7	183	# characters in file name
*00B8	184	Current logical file
*00B9	185	Current secondary address
*00BA	186	Current device
*00BB-00BC	187-188	Pointer: to file name
00BD	189	Write shift word/Read input char
00BE	190	# blocks remaining to Write/Read
00BF	191	Serial word buffer
00C0	192	Tape motor interlock
00C1-00C2	193-194	I/O start addresses
00C3-00C4	195-196	KERNAL setup pointer
*00C5	197	Current key pressed
*00C6	198	# chars in keyboard buffer
*00C7	199	Screen reverse flag
00C8	200	Pointer: End-of-line for input
00C9-00CA	201-202	Input cursor log (row, column)
*00CB	203	Which key: 64 if no key
00CC	204	cursor enable (0 = flash cursor)
00CD	205	Cursor timing countdown
00CE	206	Character under cursor
00CF	207	Cursor in blink phase
00D0	208	Input from screen/from keyboard
*00D1-00D2	209-210	Pointer to screen line
*00D3	211	Position of cursor on above line
00D4	212	0 = direct cursor, else programmed
*00D5	213	Current screen line length
*00D6	214	Row where cursor lives
00D7	215	Last inkey/checksum/buffer
*00D8	216	# of INSERTs outstanding
*00D9-00F0	217-240	Screen line link table
00F1	241	Dummy screen link
00F2	242	Screen row marker
*00F3-00F4	243-244	Screen color pointer
00F5-00F6	245-246	Keyboard pointer
00F7-00F8	247-248	RS-232 Rcv pointer
00F9-00FA	249-250	RS-232 Tx pointer

* *Useful memory location*

HEX	DECIMAL	DESCRIPTION
*00FB-00FE	251-254	Operating system free zero page space
00FF	255	Basic storage
0100-010A	256-266	Floating to ASCII work area
0100-013E	256-318	Tape error log
0100-01FF	256-511	Processor stack area
*0200-0258	512-600	Basic input buffer
*0259-0262	601-610	Logical file table
*0263-026C	611-620	Device # table
*026D-0276	621-630	Secondary Address table
*0277-0280	631-640	Keyboard buffer
*0281-0282	641-642	Start of memory for op system
*0283-0284	643-644	Top of memory for op system
0285	645	Serial bus timeout flag
*0286	646	Current color code
0287	647	Color under cursor
*0288	648	Screen memory page
*0289	649	Max size of keyboard buffer
*028A	650	Key repeat (128 = repeat all keys)
*028B	651	Repeat speed counter
028C	652	Repeat delay counter
*028D	653	Keyboard Shift/Control flag
028E	654	Last keyboard shift pattern
028F-0290	655-656	Pointer: keyboard decode logic
*0291	657	Shift mode switch (0 = enabled, 128-locked)
0292	658	Auto scroll down flag (0 = on, <>0 = off)
0293	659	RS-232 control register
0294	660	RS-232 command register
0295-0296	661-662	Nonstandard (Bit time/2-100)
0297	663	RS-232 status register
0298	664	Number of bits to send
0299-029A	665-666	Baud rate (full) bit time
029B	667	RS-232 receive pointer
029C	668	RS-232 input pointer
029D	669	RS-232 transmit pointer
029E	670	RS-232 output pointer
029F-02A0	671-672	Holds IRQ during tape operations
02A1-02FF	673-767	Program indirects
*0300-0301	768-769	Error message link
0302-0303	770-771	Basic warm start link
0304-0305	772-773	Crunch Basic tokens link
0306-0307	774-775	Print tokens link
0308-0309	776-777	Start new Basic code link

* *Useful memory location*

HEX	DECIMAL	DESCRIPTION
030A-030B	778-779	Get arithmetic element link
030C	780	Storage for 6502 .A register
030D	781	Storage for 6502 .X register
030E	782	Storage for 6502 .Y register
030F	783	Storage for 6502 .P register
0310-0313	784-787	??
0314-0315	788-789	Hardware (IRQ) interrupt vector (EABF)
0316-0317	790-791	Break interrupt vector (FED2)
0318-0319	792-793	NMI interrupt vector (FEAD)
031A-031B	794-795	OPEN vector (F40A)
031C-031D	796-797	CLOSE vector (F34A)
031E-031F	798-799	Set-input vector (F2C7)
0320-0321	800-801	Set-output vector (F309)
0322-0323	802-803	Restore I/O vector (F3F3)
0324-0325	804-805	INPUT vector (F20E)
0326-0327	806-807	Output vector (F27A)
0328-0329	808-809	Test-STOP vector (F770)
032A-032B	810-811	GET vector (F1F5)
032C-032D	812-813	Abort I/O vector (F3EF)
032E-032F	814-815	user vector (FED2)
0330-0331	816-817	Link to load RAM (F549)
0332-0333	818-819	Link to save RAM (F685)
0334-033B	820-827	??
*003C-03FB	828-1019	Cassette buffer
0400-0FFF	1024-4095	3K expansion RAM area
1000-1DFF	4096-7679	User Basic area
1E00-1FFF	7680-8191	Screen memory
2000-3FFF	8192-16383	8K expansion RAM/ROM block 1
4000-5FFF	16384-24575	8K expansion RAM/ROM block 2
6000-7FFF	24576-32767	8K expansion RAM/ROM block 3

NOTE: When additional memory is added to block 1 (and 2 and 3), the KERNAL relocates the following things for BASIC:

1000-11FF	4096-4607	Screen memory
1200-?	4608-?	User Basic area
9400-95FF	37888 = 38399	Color RAM

8000-8FFF	32768-36863	4K Character generator ROM
8000-83FF	32768-33791	Upper case and graphics
8400-87FF	33792-33815	Reversed upper case and graphics
8800-8BFF	33816-35839	Upper and lower case
8C00-8FFF	35840-36863	Reversed upper and lower case
9000-93FF	36864-37887	I/O BLOCK O

* *Useful memory location*

HEX	DECIMAL	DESCRIPTION
9000-900F	36864-36879	Address of VIC chip registers
9000	36864	bits 0-6 horizontal centering bit 7 sets interlace scan
9001	36865	vertical centering
9002	36866	bits 0-6 set # of columns bit 7 is part of video matrix address
9003	36867	bits 1-6 set # of rows bit 0 sets 8×8 or 16×8 chars
9004	36868	TV raster beam line
9005	36869	bits 0-3 start of character memory (default = 0) bits 4-7 is rest of video address (default = F) BITS 3,2,1,0 CM starting address
9006	36870	horizontal position of light pen
9007	36871	vertical position of light pen
9008	36872	Digitized value of paddle X
9009	36873	Digitized value of paddle Y
900A	36874	Frequency for oscillator 1 (low) (on: 128-255)
900B	36875	Frequency for oscillator 2 (medium) (on: 128-255)
900C	36876	Frequency for oscillator 3 (high) on: 128-255)
900D	36877	Frequency of noise source
900E	36878	bit 0-3 sets volume of all sound bits 4-7 are auxiliary color information
900F	36879	Screen and border color register bits 4-7 select background color bits 0-2 select border color bit 3 selects inverted or normal mode

		HEX	DEC
0000	ROM	8000	32768
0001		8400	33792
0010		8800	34816
0011		8C00	35840
1000	RAM	0000	0000
1001		xxxx	
1010		xxxx	unavail.
1011		xxxx	
1100		1000	4096
1101		1400	5120
1110		1800	6144
1111		1C00	7168

HEX	DECIMAL	DESCRIPTION
9110-911F	37136-37151	6522 VIA#1
9110	37136	Port B output register (user port and RS-232 lines)

PIN ID	6522 ID	DESCRIPTION	EIA	ABV
C	PB0	Received data	(BB)	Sin
D	PB1	Request to Send	(CA)	RTS
E	PB2	Data terminal ready	(CD)	DTR
F	PB3	Ring Indicator	(CE)	RI
H	PB4	Received line signal	(CF)	DCD
J	PB5	Unassigned	( )	XXX
K	PB6	Clear to send	(CB)	CTS
L	PB7	Data set ready	(CC)	DSR
B	CB1	Interrupt for Sin	(BB)	Sin
M	CB2	Transmitted data	(BA)	Sout
A	GND	Protective ground	(AA)	GND
N	GND	Signal ground	(AB)	GND

9111	37137	Port A output register (PA0) Bit 0 = Serial CLK IN (PA1) Bit 1 = Serial DATA IN (PA2) Bit 2 = Joy 0 (PA3) Bit 3 = Joy 1 (PA4) Bit 4 = Joy 2 (PA5) Bit 5 = Lightpen/Fire button (PA6) Bit 6 = Cassette switch sense (PA7) Bit 7 = Serial ATN out
9112	37138	Data direction register B
9113	37139	Data direction register A
9114	37140	Timer 1 low byte
9115	37141	Timer 1 high byte & counter
9116	37142	Timer 1 low byte
9117	37143	Timer 1 high byte
9118	37144	Timer 2 low byte
9119	37145	Timer 2 high byte
911A	37146	Shift register
911B	37147	Auxillary control register
911C	37148	Peripheral control register (CA1, CA2, CB1, CB2) CA1 = restore key (Bit 0) CA2 = cassette motor control (Bits 1-3) CB1 = interrupt signal for received RS-232 data (Bit 4) CB2 = transmitted RS-232 data (Bits 5-7)
911D	37149	Interrupt flag register
911E	37150	Interrupt enable register
911F	37151	Port A (Sense cassette switch)
9120-912F	37152-37167	6522 VIA#2
9120	37152	Port B output register keyboard column scan (PB3) Bit 3 = cassette write line (PB7) Bit 7 = Joy 3
9121	37153	Port A output register keyboard row scan
9122	37154	Data direction register B
9123	37155	Data direction register A
9124	37156	Timer 1, low byte latch
9125	37157	Timer 1, high byte latch
9126	37158	Timer 1, low byte counter
9127	37159	Timer 1, high byte counter timer 1 is used for the 60 time/second interrupt
9128	37160	Timer 2, low byte latch
9129	37161	Timer 2, high byte latch
912A	37162	Shift register
912B	37163	Auxiliary control register
912C	37164	Peripheral control register CA1 Cassette read line (Bit 0) CA2 Serial clock out (Bits 1-3) CB1 Serial SRQ IN (Bit 4) CB2 Serial data out (Bits 5-7)
912D	37165	Interrupt flag register
912E	37166	Interrupt enable register
912F	37167	Port A output register
9400-95FF	37888-38399	location of COLOR RAM with additional RAM at blk 1
9600-97FF	38400-38911	Normal location of COLOR RAM
9800-9BFF	38912-39935	I/O block 2
9C00-9FFF	39936-40959	I/O block 3
A000-BFFF	40960-49152	8K decoded block for expansion ROM
C000-DFFF	49152-57343	8K Basic ROM
E000-FFFF	57344-65535	8K KERNAL ROM

HEX	DECIMAL	DESCRIPTION
00A0-00A2	160-162	3 byte jiffy clock. The TI and TI$ variables are translations of these locations.
00B2-00B3	178-179	Points to the start of the tape buffer. Can be used as an indirect zero-page jump to a routine in the buffer.
00B7	183	Number of characters in filename.
00B9	185	Which secondary address is currently being used.
00BA	186	Current device number being accessed.
00BB-00BC	187-188	Points to location of filename in memory.
00C5	197	Current key being held down. There will be a 64 here if nothing is held down. If more than 1 key is down, the key with the highest number on the chart is what shows up here.

#	key	#	key	#	key	#	key
0	1	16	none	32	space	48	Q
1	3	17	A	33	Z	48	E
2	5	18	D	34	C	50	T
3	7	19	G	35	B	51	U
4	9	20	J	36	M	52	O
5	+	21	L	37	•	53	@
6	£	22	;	38	none	54	↑
7	DEL	23	←CRSR→	39	f1	55	f5
8	←	24	STOP	40	none	56	2
9	W	25	none	41	S	57	4
10	R	26	X	42	F	58	6
11	Y	27	V	43	H	58	8
12	I	28	N	44	K	60	0
13	P	29	,	45	:	61	—
14	*	30	/	46	=	62	HOME
15	RETURN	31	↑CRSR↓	47	13	63	17

00C6	198	Number of characters currently in keyboard buffer.
00C7	199	Flag for reverse on/off. A 1 here is on, a 0 is off.
00CB	203	Same as 197.
00D1-00D2	209-210	Address of start of line where cursor is.
00D3	211	Position of cursor on line.
00D5	213	Current screen line length—either 21, 43, 65, or 87.

USEFUL MEMORY LOCATIONS

This is a more in-depth guide to some of the memory locations you can use.

HEX	DECIMAL	DESCRIPTION
0014-0015	20-21	Where BASIC stores integer variables used in calculations. The fixed-float and float-fixed routines (vectors at 1-2 and 3-4) use the value in this area.
002B-002C	43-44	The start of the BASIC program in memory. Location 43 contains the low byte, and location 44 has the high byte. To compute the start of BASIC in decimal, use the formula: PEEK(43) + 256 * PEEK(44)
002D-002E	45-46	The start of the numeric variables, which is usually immediately after the end of the BASIC program.
002F-0030	47-48	The start of arrays in memory, usually immediately following the numeric variables.
0031-0032	49-50	The end of the arrays in memory.
0033-0034	51-52	Bottom of string storage, moving from the top of available memory down to the top of arrays.
0037-0038	55-56	The top of free RAM. By lowering this value, some RAM can be "protected" against BASIC putting values here.
0043-0044	67-68	Jump vector for INPUT statement.
0061-0066	97-102	Floating point accumulator #1 for calculations.
0069-006E	105-110	Floating point accumulator #2.
0073-008A	115-138	The CHRGET subroutine resides here. This routine gets the next BASIC character from machine language.
0090	144	Status word ST.
0098	152	Number of open files.
0099	153	Device number for input, normally 0 (keyboard).
009A	154	Output (CMD) device, normally 3 (screen).

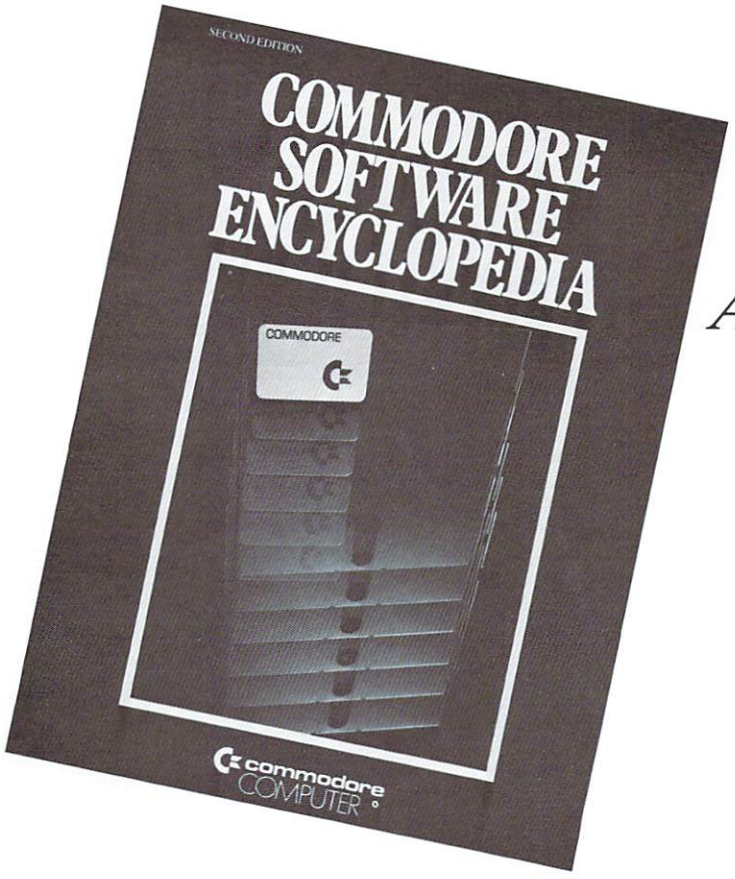
HEX	DECIMAL	DESCRIPTION
00D6	214	Screen row where cursor is. To change the cursor position, locations 201, 210, 211, and 214 must be changed.
00D8	216	Number of spaces left in INSERT mode. POKEing this to a zero will turn off insert mode.
00D9-00F0	217-240	Screen line link table. A 158 means that the line is finished at the end of that line, and a 30 means that the line continues on the next line.
00F3-00F4	243-244	Pointer to the current space in color memory.
00FB-00FE	251-254	Available locations in zero page.
0200-0258	512-600	BASIC input buffer—where the characters being INPUT will go.
0259-0262	601-610	Logic 1 file table for OPEN files.
0263-026C	611-620	Device # table for OPEN files.
026D-0276	621-630	Secondary address table
0277-0280	631-640	Keyboard buffer. If characters are POKEd in here and location 198 (# of characters in buffer) is changed, it will be as if the characters were typed from the keyboard.
0281-0282	641-642	Start of memory pointer.
0283-0284	643-644	Top of memory pointer.
0286	646	Current color code. This holds the color number that goes into color memory during PRINT operations.
0288	648	Screen memory page. If you want the operating system to know where screen memory is, this must be changed as well as the VIC chip.
0289	649	Maximum size of keyboard buffer. If this is set greater than 10, vital pointers will be destroyed.
028A	650	Keyboard repeat flag. If this is a 0, only cursor controls repeat; if 128, all keys repeat.
028B	651	This determines how long the VIC waits before repeating key.
028D	653	Keyboard SHIFT, CTRL, Commodore flag. The SHIFT sets the 1 bit, Commodore sets the 2 bit, and the CTRL sets the 4 bit.
0291	657	Setting this location to 128 will disable switching case, and a 0 here enables the ability to switch.
0300-0301	768-769	This is the jump vector for errors. By changing this vector, a routine can intercept any error condition.
033C-03FB	828-1019	Cassette buffer. This is where data files are held before they are INPUT. When not using files, this is available for POKEing or machine language programs.

Memory Map by;

Jim Butterfield
Andy Finkel
Neil Harris

Excerpt from the Programmer's Reference Giude

WE WROTE THE BOOK ON SOFTWARE!



OVER 1,000 APPLICATIONS ONLY $9.95

The Commodore Software Encyclopedia is a comprehensive directory of over 1000 programs for business, education, personal, and recreational applications. Listings are provided for such categories as:

- Business
  including Dow Jones Portfolio Management, Legal Time Accounting, Accounts Receivable, Inventory Systems, Tax Preparation, Real Estate Programs, Payroll, General Accounting, Mailing List Management, and Visicalc.
- Word Processing
- Utilities
- Engineering Aids
- Personal Aids
- Games
- VIC-20
- Publications
- Software Vendors
- Hardware Overview
- Overseas Suppliers
- Education
  including Math, Science, Social Studies, Verbal Skills, and Administration.

Whether you're interested in personal computers or own a Commodore, this directory can be an invaluable one in locating just the right software package for you.

Pick one up at your local Commodore dealer today or return this coupon.

Please send _____ Software Encyclopedia(s) @ $9.95 + 2.00 postage and handling. Enclose check or money order.

Name _____

Address _____

_____

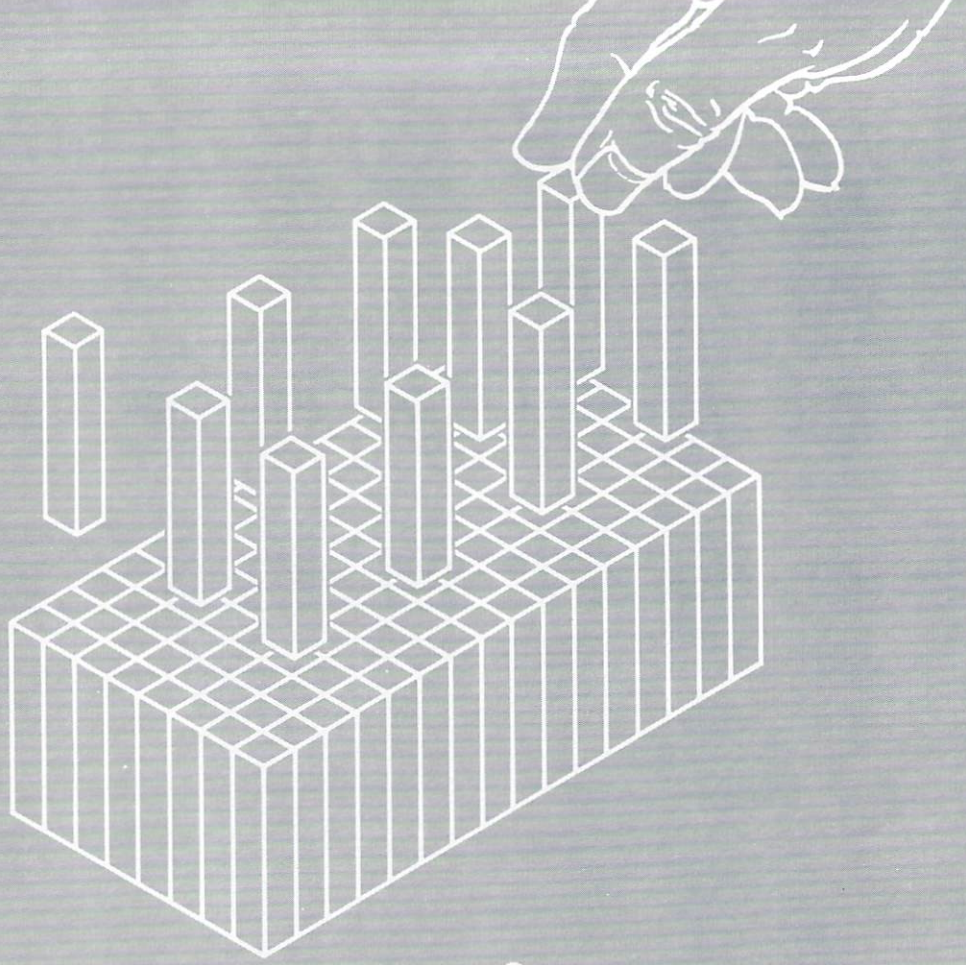
_____

_____

Return to: Commodore Magazine
Commodore Business Machines
681 Moore Road
King of Prussia, Pa. 19406

☐ I own a Commodore Computer Model _____

commodore COMPUTER



Easy Choice

The TOOLKIT™ is one of the most widely used enhancements for the PET/CBM microcomputer. This collection of programming aids makes writing, debugging and polishing BASIC programs easier - so easy that these 9 "tools" have become an integral part of the PET microcomputer.

Ask for it by name -
See your local Commodore dealer.

AUTO	your PET will automatically number lines so you don't have to manually enter this information.
DELETE	every line within the range of numbers will be instantly removed from your program without the tedious process of entering line numbers and pressing return.
HELP	simply by typing "help" you will receive immediate information as to where your error occurred.
RENUMBER	this feature allows you to change all line numbers and all references to those numbers.
TRACE	now you can see the exact order and sequence of your program's execution.
STEP	like TRACE, but one statement at a time can be executed.
DUMP	during or after running a program, your PET will display the names and values of all variables used in your program.
FIND	this command allows you to zero-in on only those lines which contain the desired character string.
APPEND	zdds a program previously saved on tape to your current program.

The original **only $39.95!**

BASIC Programmer's TOOLKIT™

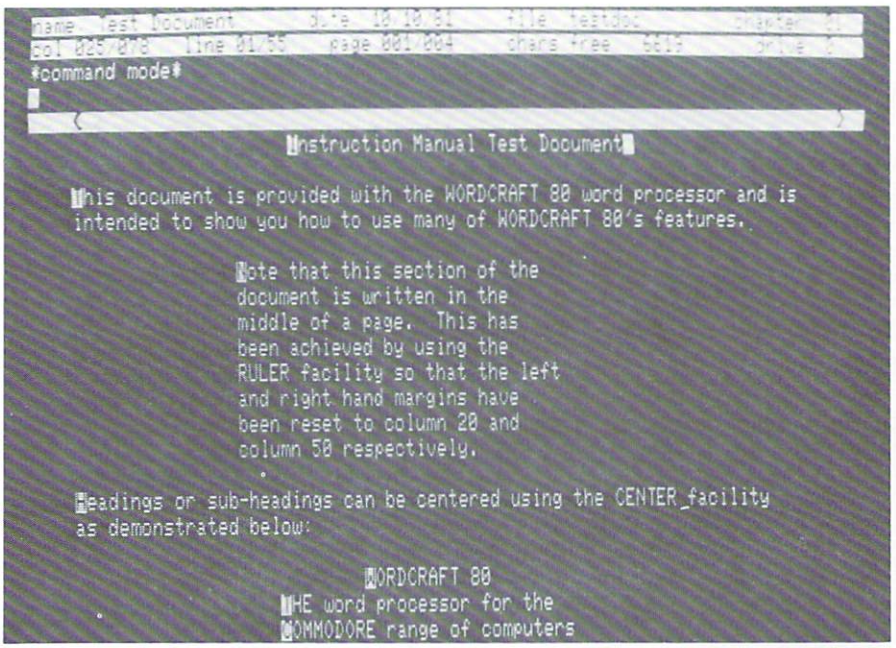
PALO ALTO IC's

2585 East Bayshore Rd., Palo Alto, California 94303 (415) 493-8665

Wordcraft 80

Wordcraft 80 is a complete word processing system designed exclusively for Commodore computers. The system sells for $395.00. Wordcraft 80 presents copy on the screen in a format virtually identical to the way in which it will be output from the printing device. Subject to the text being output in a format not exceeding 80 columns, the margins and tabulations set up by the operator directly on the screen will apply when the material is output to the printer stage. There is no intermediate reformatting stage between displaying text on input and the hard-copy output.

The only exceptions are in the process of right margin justification, the underlining of text on the screen and special functions—for example, the printing of superscripts and subscripts. There the hardware is unable to cope accurately with the display of those features.



Large formats

Large page formats are possible. Text may be sent to the printer in page units up to 117 columns wide by a maximum depth of 98 lines. When the required page area is greater than the 80 column by 20 text lines of the screen, it is achieved using automatic scrolling of text in vertical and horizontal directions.

Subject to the capacity of a text storage disk, there is no restriction on the size of a text file.

Taking into account the amount of memory required for the program and the page display, there is a capacity of approximately 10KB for the manipulation of text. For convenience of handling the text, the file—Wordcraft refers to it as a document—is broken-down into a sequence of 'chapters', none of which may exceed the 10,000-character limit. The software, nevertheless, can handle page numbering and chapter numbering throughout the length of a larger document.

Apart from information held in the display area, text is held in the memory in a packed format, rather than the layout in which it is presented on the screen. While that makes more effective use of the memory space—unnecessary blank spaces do not have to be held as finite characters in the memory, for example—there is a potential penalty when attempting to process text rapidly. In any event, the short delay which may be experienced during editing is more than compensated for by the greater storage capacity that is possible.

While the total depth of the screen is 25 lines, the first five are absorbed with status information. The file name and data are both displayed, along with parameters defining the position of the cursor on the screen, in terms of column and line, and the page number within the current chapter.

Count-down

With a capacity of 10,000 characters available for text storage, an essential feature of the screen layout information is a count-down of the number of characters remaining.

The lowermost of the five display lines is a 'ruler' which shows the current position of the left margin, the right margin, tab settings and forced indentations.

Wordcraft 80 is designed to function in one of three modes—Command, Type and Control. The current mode is displayed in line four of the display. **Command** is called-up when the system is loaded and the printer output has been selected. As an aid to the operator, the current version provides a help table to ensure that the correct printer option is selected. In command mode, specific commands may be entered through the keyboard to call-up files from disk, issue the →

printing instructions and control justification.

Since the cursor is pre-set at the start of the mode line—line four in the display area—the commands entered through the keyboard will appear in the display and remain there until executed.

Logical

The command mode will revert to **Type**, the second of the operational modes, when the STOP key is pressed. That allows text to be keyed into the lower 20 lines of the screen. Striking STOP again while the system is accepting text will return the system to command mode for the input of further instructions.

So far as possible, all commands have been structured in an admirably logical manner. The entry required to call-up ('get') chapter 3 of a file from a disc in drive 0 would be **g, filename,0,3.** The corresponding instruction for the relatively complex task of merging pages four to seven of a file held in drive 1 into the current document would be **m,filename,1,4-7.**

Once the basic principles of command structure have been learned, the use of even the most powerful command should be within the grasp of the operator.

In **Control,** the third of the operating modes, Wordcraft will accept editing instructions for manipulating the text already within the memory. Transferring from the type mode to the control mode requires a single key-stroke on the RVS key. Only the most basic editing functions can be performed without the use of this control mode. Those are the over-typing of text on the screen for character-for-character replacement, the deletion of individual characters (DEL) and single insertions (INS).

Editing

When editing text in the type mode, the DEL key will remove the character at the point of the cursor, a logical approach to text correction. More extensive deletions require the control mode RVS. DEL will remove the complete line of text in which the cursor is then situated. That approach is less satisfactory than one which deletes text only from the point of the cursor to the end of the current line.

The use of commands for deletion is an area fraught with problems for the operator on many word processing systems. Frequently there is insufficient security to prevent the accidental erasure of large volumes of text.

In the Wordcraft 80 package, the effect of DEL is instantaneous; the characters removed in that way cannot be recovered on the screen.

The command to erase larger blocks of text provides protection, however, and is one of three closely-related procedures for moving, copying and deleting text within the confines of a section. Provided a specific introductory command is required—RVS e, for example—followed by a shifted RVS keystroke at the end of the block, there is adequate security to prevent the accidental erasure of significant amounts of text.

The original text will not have been deleted from the copy of the file on disk until the amended version has been copied back from the memory so, in theory, there is a backup of the job on the floppy disk until the re-write command has been issued.

The deletion might have been only one of many intricate editings required in a long and complex document and to have to call-up the deleted text from the source document to preserve the other changes to the text would be unnecessary and time-consuming.

In all, the checks on an operator deleting text accidentally are sufficient.

The flexibility of software design is tested to the fullest when the system is required to search for a string of text and replace it. Identifying changes of character case is perhaps the greatest problem for many word processors—searching for 'computer' would not identify any occurrences of 'Computer' and 'COMPUTER'.

Even more important is the way in which the program replaces strings. Would an exchange string entered as lower-case characters by the operator take upper-case when the change was made at the beginning of a new sentence?

Wordcraft 80 has the abilty to recognize a string of characers irrespective of the style in which it appears on the screen. The replacement facility operates on two levels, so that the system can cope with individual—'local'—exchanges or throughout the length of a document—'global inerchange'. Adjustments may be made automatically for the case of the inserted string if that is required,so the operator may then call-up a global interchange without having to monitor the first letter of each replacement.

Adequate warning

A potential problem would be where a short string has to be replaced by a block of text several times its own length. There is a possibility that the memory might be insufficient to accommodate the revised version of the chapter but Wordcraft 80 provides an adequate warning when the memory limit is approached.

Since text is displayed on the screen in exactly the same layout as it will appear on the printed page, there is no intermediate processing or display stage between the entry of the text through the keyboard and the output of the material as hard copy.

Print commands adopt a similar structure to those for calling-up documents from the disk. Printing page 5

to 9 of the current document chapter, for example, would require a command of the form **p,5-9,** followed by a return stroke.

Wordcraft 80 provides three options for printing. First, there is a double-line copy appropriate for drafting; a character 'd' immediately after the command indicates that is required. The software allows for four line-spacings and four pitches.

Highlighting text—through emboldening, for example—is the second option, using control characters RVS (and RVS) to mark the start and finish of the highlighted section.

The third print command option allows multiple copies—up to 127—of pages to be generated automatically. Any combination of print command may be strung together so that the instruction to print four copies of pages 3 to 8 in double-line spacing would be **p,3-8,d-4.**

Right-margin justification is achieved using even white spacing between words, an effect which can produce disconcerting results on narrow columns.

Standard letters

The output of standard letters is a task which many WP systems fail to handle with anything approaching simplicity. Address files may have to be structured rigidly, so that personalized details can be merged into the text of the letter only in the same sequence in which they were recorded in the master address file. If there is an error in constructing the address file—typically omitting one of the parameters in an address—the whole of the standard letter run falls out of logical sequence.

Wordcraft 80 has opted for an alternative approach, using a field file which may contain up to 15 parameters, each identified separately. The standard letter 'blank' carries embedded 'take' codes which allow any number of those parameters to be called-up from the address file. The benefit of the method is that a master file can be developed for the mutual benefit of sales departments, accounts and service organizations, calling the appropriate details into standard output as necessary.

Conclusions

- Wordcraft 80 has taken full advantage of the 80-column industry-standard screen, allowing the operator to display text at the input stage exactly as it will appear on output.
- A software house designing a word processing package for the office market must take into account the needs and conventions of a secretary. To this end, Wordcraft 80 is more acceptable as a typing system than many competitive systems.
- Wordcraft 80 is a more logical package for a relatively-inexperienced operator to handle.
- Since standardized correspondence is one attraction of office word processing systems for many users, the Wordcraft 80 package has great flexibility to select specific data from a master file. A single data file could be prepared for use on the Wordcraft system, which could be accessed by every user department within an organization■

PRODUCT REVIEW

POWER

by
Elizabeth Deal

POWER is a ROM chip for the PET/CBM computers that adds new commands to BASIC. Versions are available for the Upgrade PET/CBMs and Basic 4—40 and 80 column computers. It can be used with disk or tape. It does not affect the speed of the BASIC program being executed. It can be turned off when not needed. It turns some commands off when they should be off. Programs written using Power run correctly on systems not equipped with Power.

Power is distributed worldwide by Professional Software Inc. The price is $89. Installation is into the UD3 socket on the 40-col 9" screen computers, and UD12 on the 40 or 80-col 12" screen machines. The address of Power is $9000 or 36864.

The program was written by Brad Templeton. Jim Butterfield wrote the instructions. The Power chip contains several commands unheard of in small systems. The manual contains vital information that rarely is provided. It includes interesting explanations of how Power and PET function together.

Power is not a run-of-the-mill BASIC extender. It contains well thought out solutions to working with BASIC, and permits customizing your PET to your needs. It even permits adding commands of your choice, the design of this feature being meticulously thought out.

There is little in the way of hard coding. Programs can be relocated, and loaded into areas other than the usual 1025. Every Power command that works with BASIC in its usual location can be used in any other position. Programs can be saved as ASCII files, which allows saving the entire text or any section of it. Power makes no assumptions about device numbers in your system or your general system configuration. Power helps, but does not get in the way.

Power is user-oriented. It seems to have been designed from lots of experience in debugging. One of the first features you notice when beginning to use Power is that it treats a program in memory as it it were on a continuous belt. The program can be scrolled on the screen by use of the up and down cursor keys. I haven't used a LIST command since I got Power. The entire program can be scrolled, and when it reaches the end, the beginning reappears. If a program is long, then, of course, you may see a section by typing a number and holding the down-cursor key. It's a marvelous feature that makes use of the PET easier than ever before.

Correcting programs is a snap as the cursor and INST/DEL keys repeat.

Changing programs is facilitated by the search (@) and replace (]) commands. So what, I hear you say, my BASIC Aid has that too. This is better. Suppose you're looking for a variable X$. You can ask for all or a range of lines containing X$. Or you can slowly look for each occurrence, in which case one line will show up. Then, and this is the neat trick, by pressing one key, subsequent lines are displayed. Until changed by the user, the program remembers what it was looking for.

In a similar fashion a variable can be changed to another variable. You'd say to Power: replace all X$ by AB$. Also, replacement can be in a range of lines or, line by line, by pressing one key, once the initial "change" definition has been issued.

As if this weren't enough, both search and replace are intelligent commands. By use of several metacharacters, or defining characters, you may conduct a meaningful search for keywords, variables or text occurring only in a specified place. You may also search for sequences of text, as in "find all occurrences of FORNEXT". In the last instance, Power will look for all lines containing the words FOR and NEXT, disregarding what lies in between.

The program that's on the belt can be stretched to accommodate expansion. That's done by renumbering. You can renumber the entire program, or you can renumber any section of the program. Hence, if you would like to insert a subroutine or several DATA lines from tape or disk, or if you simply need a bit of room for a few lines of new coding, you are no longer limited by renumbering a whole program. I really like this feature. When debugging a long program I get used to routines having their own line numbers. There is nothing more disconcerting than having numbers changed when you run out of available line numbers. Within reasonable limits, Power permits you to keep things where you want them. It invites neatness.

The XEC command permits you to merge programs together. New program lines can be brought into any place within a program in memory. XEC command reads a tape or disk file, places the new lines on the screen and into the desired location in the existing program, in the same way as if you had typed from the keyboard. Again, there is more. XEC allows loading a program into any place in the PET without resorting to any tricks. All this is clever, useful and easy to do.

XEC command is not limited to merging programs together. Anything that can be done in direct mode from the keyboard can be placed on file and automa- →

tically executed when brought back to screen. This is a marvelous feature for repetitive sequences of commands.

As you can see, the physical management of program organization has been greatly enhanced by use of Power. I often felt stifled by the limitation of line numbers. This is no longer a problem.

We all make mistakes in writing code, as both typos and logical errors crop up. Power to the rescue. The DUMP command displays all functions, single variables and their values on the screen (see instant phrase 14, below, for dump of arrays). The WHY command puts a marker on the last executed command before an error occurred, telling you where the bug is. Both commands are routine and need no more description. To ease the typing job, Power provides AUTOmatic line numbering and one keystroke entry of BASIC keywords.

Power includes a TRACE command. There is nothing routine about it. It is invaluable in detecting both typos and errors in logic. There are several types of TRACE permitted, from the simplest display of line numbers being executed to the fanciest display, on top of the screen or line number, of statements and values of the most recent variable, including the results of logical tests used in IF-THEN statements. Alternatively, the information can be seen on a scrolling screen with the statements, values, and output all mixed up. Depending on your need, there is always one way of tracing that will prove more useful than another.

Users famiiar with Brett Butler's TRACE (COMPUTE! issue #1) will know what that command can do. Brad Templeton expanded on the concept by providing three general types of TRACE plus an easier, three-way speed control. You may now execute BASIC in a single-statement at a time fashion, or as fast as you want. TRACE is a Power command easily accessible from within a BASIC program. It can be turned on and off by an SYS statement, a necessary feature if a GET statement is used.

TRACE should prove invaluable not only as a debugging aid, but as an educational aid. Since you can actually see BASIC executing in slow motion, the mysteries of a computer's number crunching process can be explained by a simple visual demonstraion. It's a nice show to watch.

The FIX command is used to reset BASIC pointers following a BASIC LOAD of programs to weird places, such as the tape buffer, which, many of you may know, can damage BASIC pointers with tragic consequences. I find it particularly useful when working in partitions, i.e. when BASIC is not at $0400, but at some higher location.

MLM command CALLs the monitor — handy if you don't like typing SYS4 or SYS1024; necessary, if you have BASIC 4 and must use monitor CALL instead of BRK instruction to leave CMD in force; desirable if you want to track subroutine or loop status on the stack, and continue a program, since a BRK command wipes out the evidence. A nice touch.

For technically oriented people, the above commands, except program scrolling and repeating cursor keys, have been patched onto, or wedged into, the CHRGET routine. They are enabled whenever Power is enabled. Scrolling, power cursor keys and the commands that follow are patched onto the IRQ system.

The reason I mention how it is done, is that usually when the IRQ system is interfered with, input and output can't take place. Power is smart. When Power sees RUN, OPEN, LOAD, and other I/O statements, it disconnnects itself. It is completely transparent to the user. It is a simple thing to reconnect those features—any BASIC or Power command, or just hit a RETURN key to turn the IRQ driven features back on again.

The additional IRQ driven features are called "instant key" macros. There are three levels of complexity: (1) instant keywords, (2) instant phrases, (3) instant subroutines, the latter two being a monument to creative laziness. The user controls which of the three levels can be active at one time, or what combinations of the three can be used to advantage in a particular situation.

(1) "Instant keywords" is the simplest level. When enabled by the user, typing a shifted key puts in an entire keyword on the screen. For instance, shifted-I puts a word INPUT on the screen. It makes typing a program easy.

(2) "Instant phrases" is the next level, and, at least with me, seems to get the most use. You can define a shifted key to mean any sequence of commands you need. It is done by writing a REM line, which becomes part of your BASIC text, but is not executed and will not interfere in running a program on POWER-less systems.

```
14 REM"A=FORJ=ATOB:PRINTA(J);:NEXT
15 REM"O=OPEN6,4:PRINT#6,CHR$(147);:
   CMD6:LIST
```

Magic: whenever I want a listing I press shifted-O, the word OPEN and all words following it are written on the screen, I hit RETURN and the printer prints. Anything can be done in this way. That's power!

Even though Power has been designed for use in BASIC, the instant key feature can be used in the Monitor. Hence, by pressing a REM-defined key, followed by RETURN, you may modify memory, save programs, look at hunks of memory without repetitive typing of addresses, restore user commands vector following FIX, etc. Tedious monitor typing has suddenly become effortless.

It amazes me how smoothly such things work and how well Power fits in with such programs as the Supermon, Extramon, an assembler I have, and Instrument Synthesis program — all machine code. It's been programmed so carefully that nothing gets in the way. **Caution: the manual makes no claims that Power can co-exist with all machine code programs. I am just reporting what I have been able to get out of it.**

(3) Instant subroutines are used when commands can't fit on one line, or when you want to use statements not permitted in direct mode. In this case REM macros define shifted keys to mean "perform a subroutine coded in line xxxx", thus:

21 REM"D9500

What is in and following line 9500 can be any amount of code, ending with a RETURN statement. When shifted D is pressed the PET automatically executes that code. This is very powerful. It can be used for all sorts of utilites, including disk utilities in case of Upgrade systems, converting numbers from decimal to hex and back, and so on. You can customize the REM macros to a particular debugging situation you are faced with. Keep in mind that you need not save your "debugging" utilities each time you edit the program. They are there to help you write a better program; but in no way do they interfere (or waste time in case of tape systems).

Once again, these features do disconnect themselves for a program RUN, and for any direct mode I/O statements. You can also selectively disable them, if you wish.

The following may affect users of the Upgrade system: due to addressing conflict, the DOS support (Wedge) is disabled, however, a one-key "instant subroutine" replacement code is in the book.

Furthermore, here is yet another incredible feature of Power: you may add commands to Power, since the program provides a place and a careful method to hook up your own things. Therefore, the Wedge can, in fact, be reconnected and does work smoothly with Power enabled.

Additional commands can be included in the same fashion. There seems to be no limit to their number or complexity.

The documentation provided with Power is superb. Professional Software Inc. should be congratulated for putting out such a nice book, both in what it says and how it looks. The book is bound in a three ring binder, the paper is solid, unlikely to fall apart in heavy use. The whole thing is neat and professional looking.

There is a table of contents and a useful index. I make a point of it, because I have manuals that have an index so poor (if one exists at all), that it makes me wonder if it really belongs to the book to which it is physically connected, and if the author ever had to use the index to look up something. The index in the Power manual does, in fact, contain relevant references. In addition, each page is marked on top with the chapter of which it is a part, for instance "6:FINDING/CHANGING TEXT". This is handy.

The book was written by Jim Butterfield after his use of the POWER chip for over a year. Once again, it is interesting to contrast this approach to the documentation written by a programmer who knows too much to convey what is needed, or by another person who writes the instructions from the spec sheets. It has been said that; "The greatest barrier to communication is the illusion that it has been achieved." Butterfield doesn't suffer from that illusion, Power users won't either.

The Power manual is clear, concise and correct (a few typos and possibly one easy to spot bug). It is delightful to read — a mix of fun and coherent information about PET and Power. The commands are explained in simple language with numerous examples. One or two forms of explanation are given so you're bound to understand them. In several instances the syntax of commands is compared to the syntax used in similar chips, so that when you get Power, you'll clearly know what to do differently.

In the same vein, the manual contains something unusual — a list of things that might go wrong in the PET-Power interaction. Since Power adds to the invariant, existing, configuration, occasionally strange things may pop up. There is nothing unusual in that. What is unusual is to be told exactly what to expect and how and why it happens. This makes the book a strong and useful reference.

One of the unique features of the book is the documentation of the Power code itself. Much of the information is quite technical and intended for people with knowledge of the PET system and machine code. But even if you don't know much machine code, it is worthwhile to read this section. There is much to be learned there, a necessary knowledge if you plan to expand your PET's capabilities even further. First, the key addresses where such expansion can be made are listed and several "how to" hints are provided. Secondly, some of the code in Power has been written as subroutines. Those subroutines are documented in the book .What they need and what they return is clearly listed, offering a gold mine of useful utlities to be used in programs.

Finally, just as Power has a nice, comfortable feeling about it, where everything you do seems natural, where the order of parameters follows your way of thinking, so this book has a nice style to it. It describes technical things, but doesn't have a technical feel about it. It's precise without much jargon. It is funny and witty. It's a pleasure to read and have around. ■

SOFTWARE • HARDWARE FOR THE VIC 20®

COMMUNICATIONS AND COMPUTER TOOLS

- **UMI RS232 COMMUNICATOR INTERFACE** **$49.95**
  Our RS-232 Communicator Interface enables the VIC-20 to talk to most RS232 devices.
- **VICTERM A** **$19.95**
  With the VICTERM A program, begin telephone communications.
- **UMI 3K RAM CARTRIDGE** **$39.95**
  The easiest way to have 6655 bytes of program memory in your VIC.
- **UMI 3K RAM EXPANDER** **$79.95**
  For programmers, ROM developers and advanced users, our 3K RAM EXPANDER adds two ROM slots and 3K of RAM memory to the Commodore VIC-20 to give a total of 6655 bytes of available user memory. An 8-position dipswitch provides switch - selectable base address and single-socket enable/disable to two ROM sockets.
- **UMI 8K RAM CARTRIDGE** **$89.95**
  Our 8K RAM CARTRIDGE adds 8K of user memory to the VIC-20.
- **UMI SOCKETED 8K RAM/ROM BOARD** **$29.95**
  Our 8K board has four sockets for mixing and matching RAMS, ROMS, or EPROMS.
- **BUTI** (BASIC PROGRAMMER'S UTILITY ROM) **$34.95**
  A beauty of a ROM that plugs into CVH-0002, our 3K RAM EXPANDER, or CVU-0006, our RAM/ROM/EPROM Board.
- **UMI VIC-20 REFERENCE CARD** **$4.95**
  Our 8-1/2" by 11" laminated reference card is packed with commonly needed Commodore VIC20 information.

PROGRAMS FOR THE COMMODORE PET®

- **SATELLITES AND METEORITES** **$49.95**
  Satellites and Meterorites is a greater challenge than most space pilots have met.
- **SKYMATH** **$12.95**
  Decimal addition and subtraction with super PET graphics!
- **MATH SERIES DISKETTE I** **$29.95**
  Pedagogically correct programs SKYMATH, SPACEDIV, and LONGDIV, on a 5-1/4" Diskette for Commodore Disk Drives.
- **SPACEDIV** **$12.95**
  Division practice with graphics that hold the child's interest.
- **LONGDIV** **$14.95**
  20 problems of step-by-step long division with super graphics and 1-inch high digits.
- **SUPERHANGMAN** **$16.95**
  Superior graphic version of the popular spelling game.

GAMES AND ENTERTAINMENT

- **KIDDIE PAK I** **NOW $39.95**
  The KIDDIE PAK I includes the following games:
  KIDDIE CHECKERS **(SEPARATE PRICE $7.95)**
  Checkers for very small children.
  SHAPE MATCHER **(SEPARATE PRICE $11.95)**
  Fun for very small children, matching hidden patterns.
  DOGGIE MAZE **(SEPARATE PRICE $11.95)**
  A dog's race against time to reach the hydrant.
  FUN TENSES **(SEPARATE PRICE $11.95)**
  Nonsense sentences which aid children.
- **SUPER FOUR I** **NOW $49.95**
  SUPER FOUR I includes the following games:
  3-D MAZE **(SEPARATE PRICE $14.95)**
  Choose a 4 to 9 level maze, then find your way down the hallways.
  BREAKOUT **(SEPARATE PRICE $14.95)**
  No program library is complete without *BREAKOUT!*
  CAROM **(SEPARATE PRICE $14.95)**
  Planning and skillful play with deflector zaps the target.
  RACEWAY **(SEPARATE PRICE $14.95)**
  Beat the competition with a high-speed race car.
- **FABULOUS FOUR I** **NOW $59.95**
  Fabulous Four I contains the following games:
  STAR WARS **(SEPARATE PRICE $16.95)**
  Blast the tie fighter!
  DRAGON MAZE **(SEPARATE PRICE $16.95)**
  Lurking man eating dragon.
  LASER WAR **(SEPARATE PRICE $18.95)**
  Blast away space debris and aliens.
  INVADER FALL **(SEPARATE PRICE $21.95)**
  Alien paratroopers descend from space. (Requires 3K Expn.)
- **SUPER STARS I** **NOW $69.95**
  SUPER STARS I includes the following games:
  THE ALIEN **(SEPARATE PRICE $24.95)**
  Fast work with inflatable alien traps. (Requires 3K Expn.)
  AMOK **(SEPARATE PRICE $18.95)**
  Killer robots with one instruction - get the intruder.
  ALIEN BLITZ **(SEPARATE PRICE $24.95)**
  Split second reactions at the 9th level of ALIEN BLITZ will find how good you are.
  GLOBBLER **(SEPARATE PRICE $24.95)**
  The Conqueror Worm gobbles globs under your control.

ORDER NOW WHILE SUPPLIES LAST!

EDUCATION

- **SUPER ADDITION, SUBTRACTION, DON'T FALL** **$14.95**
  One cassette contains all three educational games.
- **SKYMATH** **$14.95**
  With excellent graphics, SKYMATH shoots 5-digit, 3-place addition and subtraction problems into screen's "sky" in 1-inch high digits.
- **SPACEDIV** **$14.95**
  Division practice with graphics that hold the child's interest.

GAME PROGRAM CARTRIDGES NOW AVAILABLE—ORDER NOW!

- **DEALER INQUIRIES INVITED**
- **ATTENTION SOFTWARE DEVELOPERS**
  Please contact UMI for distribution and top royalties. **Blank Cartridge kits available in quantity.**
- **MASTERCARD/VISA Accepted**

Clip Coupon for FREE CATALOG

united microware industries inc.
3431 H Pomona Blvd
Pomona, CA 91768

Please send me my FREE CATALOG describing your Hardware and Software Products

NAME _____

ADDRESS _____

CITY _____ STATE _____

ZIP _____ COMPANY _____

united microware industries, inc.

• 3431 H POMONA BLVD. • POMONA, CA 91768 • PHONE (714) 594-1351 •

VIC 20 and PET are Registered Trademarks of Commodore Business Machines

BUTTERFIELD ON COMMODORE

Half a Dialogue—Reading Keys

We've already discussed the INPUT statement in the December issue. When you do an INPUT, the program pauses and waits for the user to compose a line on the screen. When the user presses RETURN, the program resumes and uses the information entered.

This is often useful and convenient; but when we use INPUT, we don't have complete control over the user. If the user doesn't answer, the program is stopped forever, and other jobs will not take place. The user might also do undesirable things like clearing the screen, and might even stop the program if he presses RETURN without any input on the screen.

We can deal with the user on a more elemental level by using the GET command.

GET

GET takes one character directly from the keyboard buffer; the character does not go via the screen. It's usually a good idea to echo the character to the screen so that the user can see what he or she is typing (GET X$:PRINT X$;). There is a GET numeric (GET X) which gets a single numeric digit, but it's rare since the program will stop if the user inadvertantly presses an alphabetic key.

GET doesn't wait. If there's no character in the input buffer, GET returns with a null string. We can wait for a key to be pressed with a line like:

300 GET X$:IF X$="" GOTO 300

You can see that if we get no character, we go back and try again. More sophisticted versions of the same program might allow us to wait for up to 10 seconds for the user to type a key.

GET receives everything typed at the keyboard. Even cursor movements or insert and delete keys are received as single character strings. The RUN/STOP key and the SHIFT are about the only keys that GET won't receive directly.

Screen control keys—cursor move, reverse, home, etc.—are picked up directly by GET and don't influence the screen when typed. If you want them actioned, you'll have to arrange for it yourself, again by echoing the character with PRINT. On the other hand, GET is an excellent way to prevent a user from clearing the screen or doing other things that you don't want. The easiest way to identify such characters is by their ASC ASCII value, but the obvious also works: GET X$:IF X$="[HOME]" GOTO ... The Reverse-S symbol will appear where I have typed [HOME].

Sometimes there are left-over characters in the keyboard buffer. The user might have touched the keyboard accidentally, or the last key pressed might might have "bounced" and been registered twice. You can strip out such characters with simple coding like GET X$,X$,X$,X$. If the keyboard buffer contains up to four characters, they will be cleared out; if there were none, GET still doesn't hold anything up.

Remember that GET takes characters from the keyboard buffer. For one key depression, no matter if you tap a key quickly or hold it down for five minutes, only one character will go into the buffer and GET will find it there only once.

PEEK

The value of PEEK(151) will tell you whether or not a key is being held down. If you find 255 there, no key is being pressed—except maybe the SHIFT key which doesn't register there. If there is any value other than 255 in PEEK(151), somebody's holding down a key.

Special note: for Original ROMPETs, the place to check is PEEK (515). And on the VIC, check location PEEK (197); a value of 64 means that no key is being pressed.

It's possible to figure out which key is pressed based on the value you find in the PEEK location, but I don't recommend it. Different keyboards are "decoded" in different ways, and what works on one machine won't necessarily work on another. The best way to sort out which key is pressed is to use the PEEK together with the GET statement.

The trick is this: if GET says that there is no character in the keyboard buffer and PEEK says that someone is holding a key down, it's safe to assume that the key being held down is the last one you received with GET. Timing is important here, since a key could be touched in the split second between two BASIC statements. I recomend the following kind of sequence:

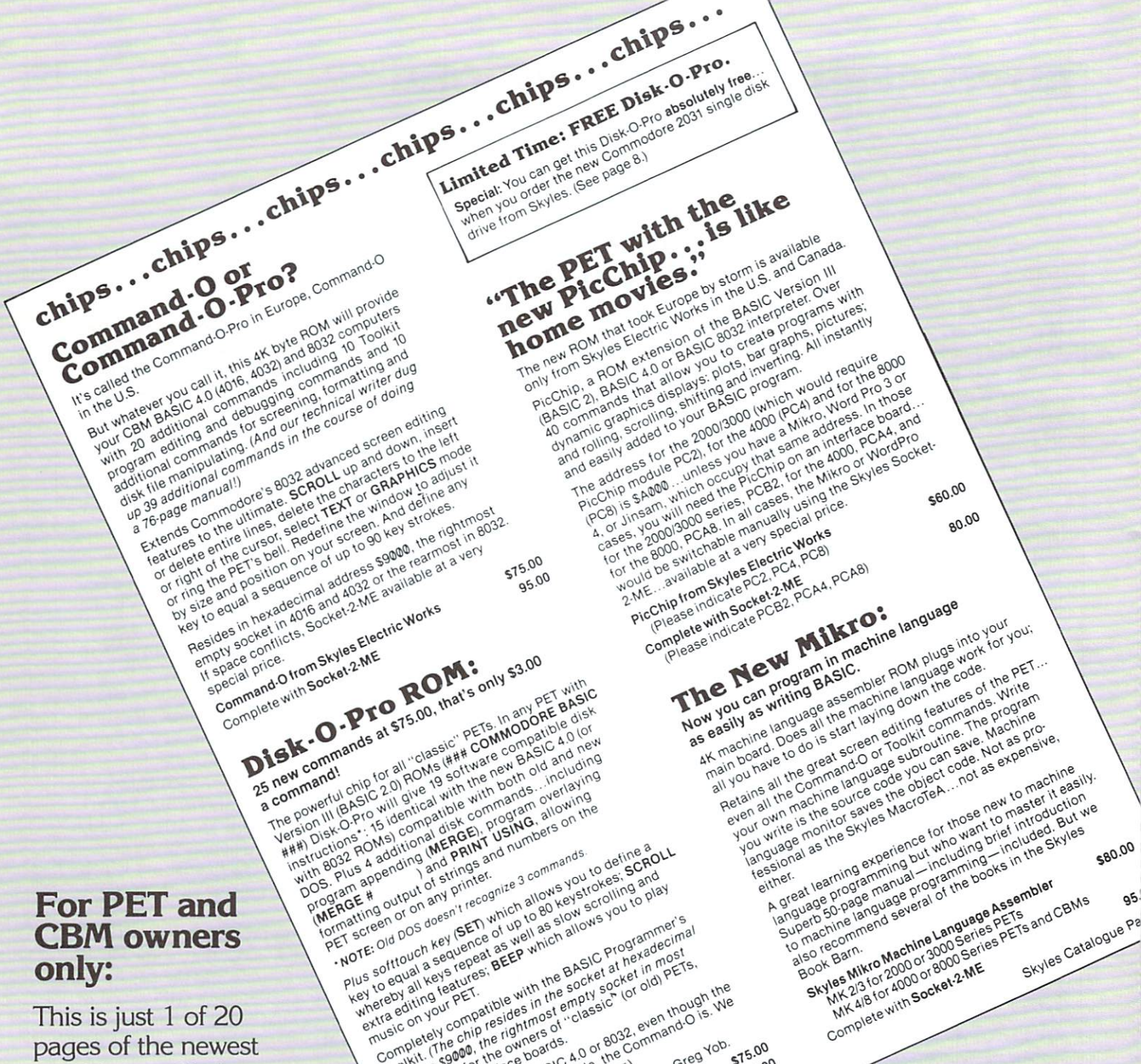
```
300 X=PEEK(151)
310 GET X$:IF X$<>""
    THEN X1×ASC(X$):GOTO 330
320 IF X=255 GOTO [. . . NO KEY ACTIVE]
330 . . . KEY ACTIVITY
```

This kind of test is very good for movement games, where you are directing something (a ball, a paddle, a tank) around the screen based on whether a key is held down or not.

Summary

GET is more elementary than INPUT. You'll need to do more work with GET, but you'll have more control over the user input.

Use the PEEK where it's necessary to find out if a key is being held down or not. It can give you a nice interface, especially where the user would otherwise pound repeatedly on a key.■



For PET and CBM owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 8½ x 10 splendor, and another 19 pages of peripherals, software and books that will make your PET or CBM computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998** (unless you live in California, in which case call **(415) 965-1735.**

From Skyles Electric Works, the oldest and largest professional specialists in the business.

Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041

PROJECTIONS & REFLECTIONS

Happy New Year!

Well it's that time again when we renew and realign our resolutions for the new year. The Software Department avoided the rush this year and set out a new game plan during December, so we are well into the grind already.

Now, in the spirit of the new year let's look into some real PROJECTIONS AND REFLECTIONS.

Here are some reflections you may remember...

Can anyone wire boards for a 407 anymore?
Does anyone out there code in Autocoder, or SPS anymore?
How about Fortran on the old IBM 1130?
Remember assembling a 10K operating system on the 360?
The GE635 and the GECOS operating system?
How about DOS DEBE, the 2400 tape drives, 2311 disk drive?
Did you ever toggle in the bootstrap, or a program on the PDP/8?
Remember DOS/VS, POWER, HASP, CICS, OS/MVS, TCAM, VSAM?
How about the first MINI's from DEC, Digital and HP?
Enter; Basic Four, Nixdorf, Microdata, Wang, and so on ...

The list goes on and on. The advances this industry has made since I started are mind blowing. The introduction of the microprocessor has literally changed the way people view computers. How many of you have the original KIM board? Our company has advanced the market probably more than any other manufacturer with our chip facilities at MOS and our "world-wide" base of people and experience.

Projections for the future are easy, because in this market anyone's guess is usually close, but let me try to refine "close" to 'right on.' Here are some of my future projections:

Microprocessors designed to run fast enough to measure them in MIPS.
Virtual memory measured in mega bytes.
Integrated communications to do networking, and electronic mail.
World-wide communicationi/information networks.
Application software products that combine all facets of business into one memory-resident module.
Governmental attempts at standardization of hardware and software.
The current market strata based on hardware will blur into one large market, and advertising and perceived need will dictate sales into any one home or business.
True 'home computers' that will literally control all power consumption devices in the home, as well as answer the door, let you view the morning paper, start breakfast and keep your personal diary.

These are just a few of the things that are well within the realm of possibility in the near future. I have seen enough in this industry to say that anything you can imagine is probably possible.

So, in adhering to those 1982 resolutions, remember to keep one of your most important assets—your imagination and sense of accomplishment. It's these attributes that have taken this industry from the above reflections, and will take it to my projections and beyond.

Go for it! ■

— *Paul Goheen*
Software Product Manager

DR. DALEY Introduces...

THE WIZ

Data Management System

THE WIZ is here!!!

THE WIZ, a system powerful enough to manage most of your data storage and manipulation needs - yet is easy to use. A system we are so sure of that we have an offer you can't resist. First though, let's take a look at a few of the many features of this program.

Feature	Benefit
1. ON-LINE help	At your fingertips is the equivalent of a 60 page manual. At any time the computer is waiting for a response from you, you may press the 'h' key or type 'help'. **THE WIZ** will then provide you with an explanation of the function you are working with.
2. Plotting capability	This is a feature unique to **THE WIZ**. It can produce a bar graph with up to 18 bars or a histogram with up to 100 points plotted. Graphically presented data is easy to interpret.
3. Wordpro interface	This option is standard with **THE WIZ**. With many of the competing data managers, if available, it is an extra cost option.
4. Read a sequential file	You may reorganize your files or even read sequential files generated by other data management systems.
5. Search for keywords	Here you can search for a word in ANY field in your record. It can even ignore differences due to upper case and lower case characters.
6. Constants in data entry	You may store up to three separate sets of contant fields. Each set can have as many fields as you like filled with information. Then two keystrokes will call the appropriate set.

And there is more. There is not room enough to tell you all the features in a one page ad!

SPECIAL OFFER

You will like our system. In fact we are so convinced of this, that we are going to pay you to try it! If you have another commercially available data management program, you can receive TRADE-IN credit for your purchase of **THE WIZ**. Call us on our toll free number (800) 548-3289 for our offer on your present system. Remember that this offer expires March 15, 1982.

DR. DALEY'S SOFTWARE

Water St. Darby, MT. 59829

(800) 548-3289 **(for Orders & Information)**

(406) 821-3924 **(Montana and technical assistance)**

Call between 9 a.m. and 6 p.m. Mountain time Monday through Friday

FILL OUT AND MAIL TODAY

THE MICROCOMPUTER MAGAZINE

commodore

Name _____

Address _____

City _____ State _____ Zip _____

Address Change ☐ Enter new address above & enclose present mailing label

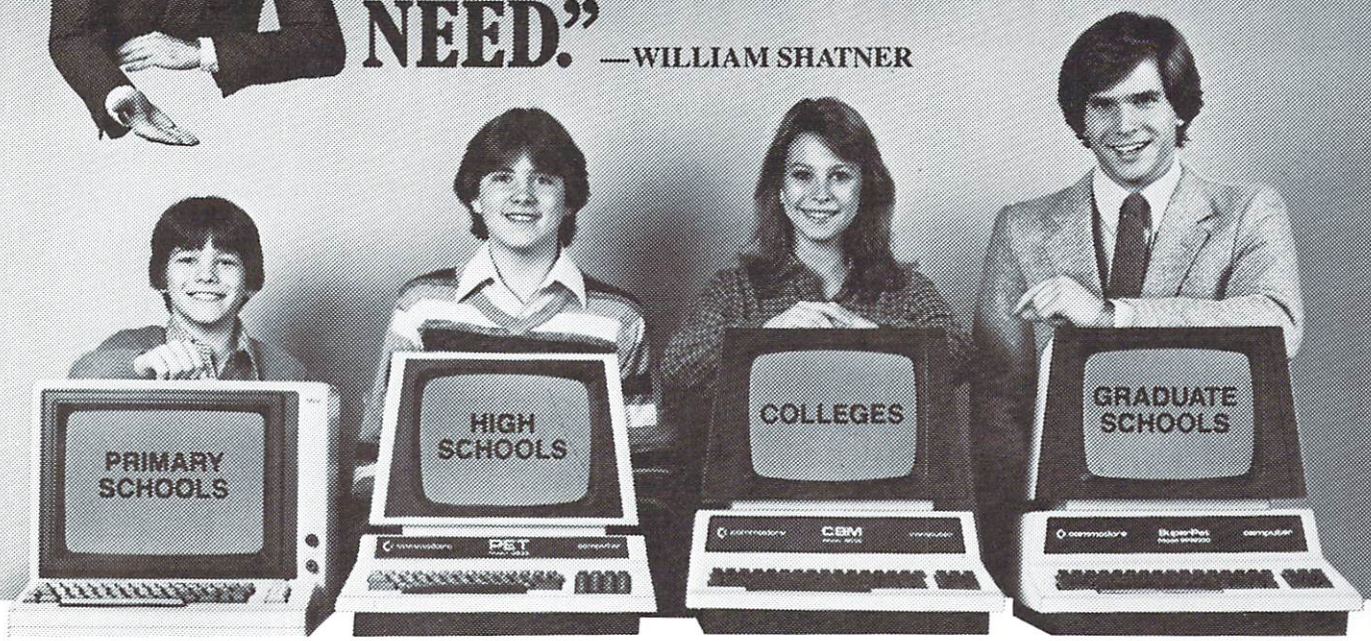
Renewal Subscription _____ New Subscription _____

Enclosed is a check or money order for $ _____ ($15.00 for six issues)
(Canadian and foreign $25.00)

for _____ issues

Make Check payable to: **COMMODORE BUSINESS MACHINES, INC.**
681 Moore Road, King of Prussia, PA 19406
Attn: Editor, Commodore Magazine

"ONLY COMMODORE HAS THE RIGHT COMPUTER FOR EVERY EDUCATIONAL NEED." —WILLIAM SHATNER



VIC-20

- So simple, a child can be running programs in minutes.
- Ideal for improving basic reading and arithmetical skills.
- Large easy-to-read characters.
- 16 screen color/4-sound generators.
- Real typewriter-style keyboard.
- Memory expandable to 32K.
- Easy-to-follow instruction guide and programming course.

$299.95

PET® 4000 Series

- World's most widely used classroom computer.
- Our lowest priced, completely self-contained computers combining keyboard, display and processor in one unit.
- Memory expandable to a full 64K.
- Built-in BASIC language.
- 128 graphic character set.
- Crisp, easy-to-read 12-inch green phosphor display.

$995

CBM™ 8032

- Students can use true business software, such as word processing, VisiCalc[1], the electronic worksheet and hundreds more.
- Professional 80-character phosphor screen.
- Easy-to-read 12″ green phosphor screen.
- Print in Basic/full screen editing capabilities.
- Business-oriented numeric keypad.
- Full 32K memory expandable to 96K.
- Built-in BASIC.

1 VisiCalc is a Trademark of Personal Software Inc.

$1495

SuperPET Computer

- Compatible with all CBM 8032 Software.
- Offers four (4) high-level languages developed by Waterloo specifically for teaching: including BASIC, APL, Pascal and FORTRAN.
- Waterloo: MicroEDITOR, 6809 Assembler & Linker programs.
- Provides 134K of memory and dual microprocessors.
- Built-in sophisticated hardware features IEEE-488 and RS-232 interfaces.
- Applications can be uploaded directly to Mainframe.
- 80 characters per line display.

$1995

Commodore's commitment can best be measured by our comprehensive range of computers, accessories, software and grant support to education.

Whatever your grade level or computer requirements there is a Commodore that will give your school cost-effective computing. From the incredible VIC-20, the only full-feature computer for under $300 to the sophisticated new SuperPET computer that offers the capabilities of a large mainframe system, at the low cost of a desktop computer.

And all Commodore computers offer the widest choice of intelligent peripherals, including printers, disk drives, cassette recorders, telephone modems and more which are compatible throughout the full range.

We wrote the book on software. Commodore's Commitment to Software... Your Commodore dealer has our new 380 page Education Resources Manual which includes four disks of 185 public domain software programs. Also, the latest edition of our Software Encyclopedia details over 1000 applications including specialized education software Math, Science, Social Studies, Verbal Skills and Administration.

Commodore's Commitment to Service... Our 600 local dealers support you and maintain your equipment. In addition a nationwide service contract is available through TRW's 200 locations for fast, reliable maintenance of your systems.

Commodore's $4.5 million Grants to Education... Commodore's Education Advisory Board granted a computer free for every two purchased on certain models. Last year this amounted to over $4.5 million.

SENSATIONAL SCHOOL OFFER:

Buy 2 PET (16K or 32K) or 2 CBM computers.
Get 1 absolutely free! Offer good for a limited time only.
SEE YOUR COMMODORE DEALER

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

Canadian Residents: Commodore Computer Systems
3370 Pharmacy Ave., Agincourt, Ont.,
Canada, M1W 2K4

☐ Please send me more information about Commodore computers for Education.

☐ I would like to subscribe to the Commodore Magazine. Enclosed is a check for $15.00.

☐ I would like a copy of the latest Software Encyclopedia. Enclosed is a check for $9.95.

Name_____ Phone_____

Grade level ____ Course _____

School name _____

School address _____

City_____ State ____ Zip ____

commodore COMPUTER

commodore
COMPUTER