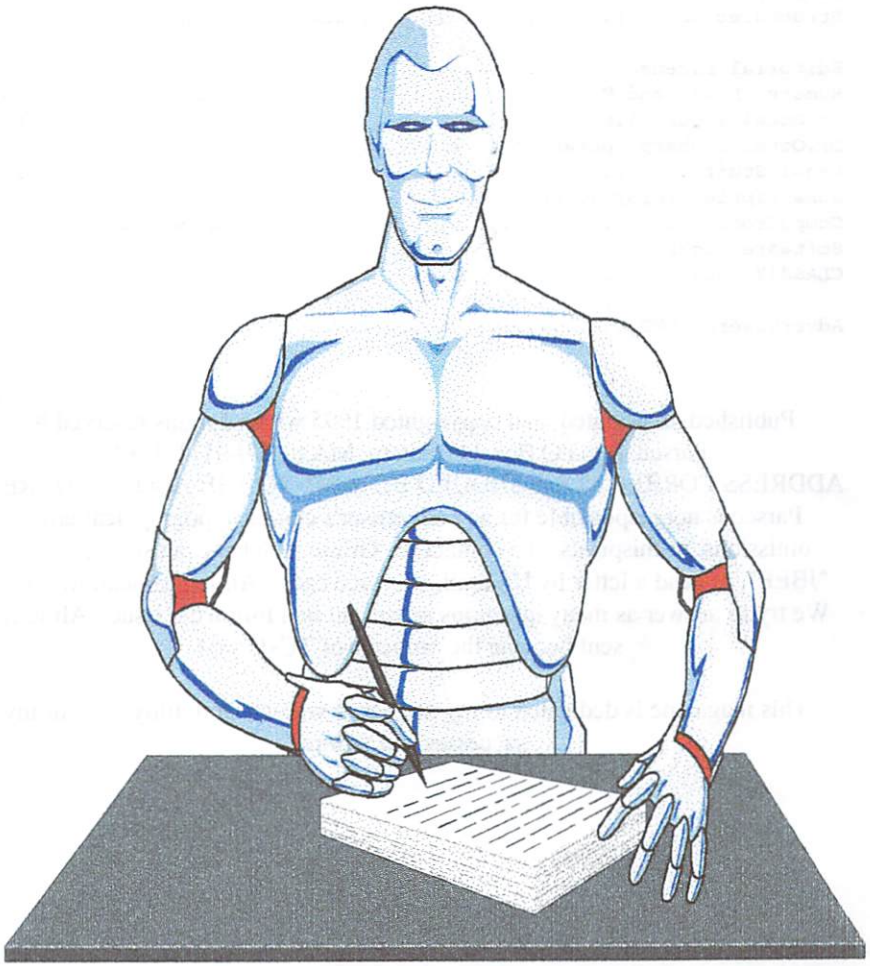
TC-128/64

The Commodore Computer Journal
Issue #37



TC-128/64 - Issue #37
The Commodore Specific Computer Journal since Jan 1986.
Publisher: Parsec, Inc.
Editor: John W. Brown

This magazine is dedicated to my mother to whom I gratefully owe all my successes and failures.

Editorial License

by John W. Brown

Greetings,

Welcome to issue #37, which is our "break even" issue. When we took over the original "stalled" TC-128, because the founder Loren Lovhaug/VMI went bankrupt, there were nearly 5,000 subscribers that were owed issues, authors owed payments, etc. Everyone that wanted their subscription has received it from Parsec. It was a long road with unexpected bumps, especially with the Commodore market taking such a dive with practically all of the vendors and magazines going out of business or dropping Commodore support altogether. Even Commodore itself went under! But, like the tortoise, we stuck it out. I just wish it had not been as slow and steady. But, the main thing is we are still here to provide support.

I would like to thank everyone that ordered products from our catalogs over the years since 1986, as it helped a great deal to fund the honoring of those issues due plus paid for the 30,000 or so free ones we gave out in the past.

As a special treat and thank you, we are putting commercial software that we bought from Kent Smotherman, offered in our catalog, on the disks for issues 37 and 38. Because I want everyone to receive the Structured BASIC programming language so you can use the short type in programs in future issues, everyone will receive a disk for issue #37. Structured BASIC 64 is too big to fit all in one issue so the manual will be split between issues #37 and #38.

Issue #38 will contain the GEOS GAME package. We are dropping the price on these two packages in our new catalog to 1/2 of what they were. Appropriate refunds for the new price differences for past customers will be contained at the start of each article and in our 1996 catalog.

As stated in one of my first editorials, we will never go out of business with your subscription money. When we do decide to close down the magazine people will be given refunds or credits towards our catalog. If you are unhappy with your subscription, you can take a credit for the undelivered issues towards products in our catalog. Which is just stating the same policy we have had.

Which brings us to Diehard magazine. It is official, they are out of business with a great amount of people owed money. The estimate from Diehard themselves is 8,000 people! I wonder what they spent all that subscription

money on from 8,000 people? I find it hard to believe that $250,000-$500,000 was wasted that fast.

I had to make up postal charts for our mailings so I decided to share them with you in time for your holiday mailings. There are no expected rate increases so these should be good for 1996 too. See side A of the companion disk.

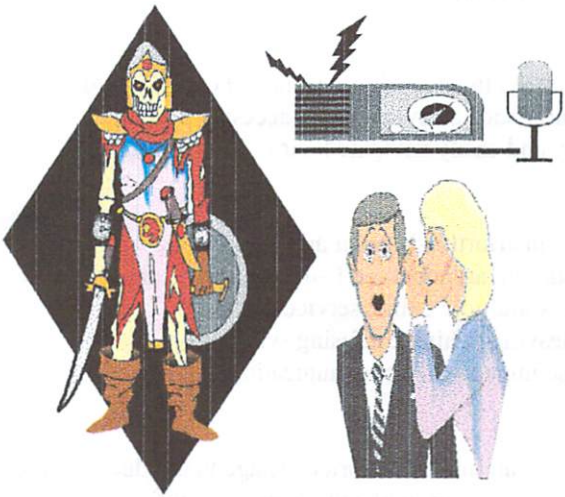
Since RIO Computers went out of business I have had inquiries about the Handyscanner CMD now sells. So included on the B side of the companion disk a review I did back in 1990. I hope this answers your questions about the Handyscanner 64.

In case you have not noticed, did I mention we have now added full color to our issues to make them spiffier and easier to read. I hope you enjoy the enhanced and new format. Another new feature is our crossword puzzle "CompuCross" which I whipped up. The questions (pages 23/26) can be found on the back of the CMD full page ad (pages 24/25) which is included as a loose folding insert.

We have also picked up the exclusive distribution rights to the Susan Lamb series of GEOS art disks. They will be offered in our new 1996 catalog along with other new products. Enjoy yourself and using the best 8 bit ever made.

Yours,

John Brown, Editor
GEnie=JBEE



News, Rumors, and Mayhem!

By John W. Brown (JBEE)

Commodore Ribbon Source

Midwest Micro 1-800-972-8822
Mon-Fri 9am-7pm est
Sat 10am-4pm est,

Commodore	Item #	6+
4022, 4512,8022	000339	$1.59
MPS 803	001903	$2.87
MPS 1200/1250	001923	$2.55

GEnie Unveils New Look, Lower Pricing, and Internet Access

Users Gain From Better Benefits, Stronger Value

Rockville, MD, April 3, 1995- - GEnie Services, a leading global provider of business and consumer online services, announced today three key improvments including a new graphical interface, the first step in a streamlined pricing structure and full internet access.

"GEnie is redefining its offering in a way that will deliver value to the end-user,"said Doug Wolford, manager of communication and public affairs, GE Information Services.

Details of the announcement include:

Redesigned Front End

Available immediately, the new GEnie front end offers a bold design, supported by creative features such as dynamically accessible icons that enable users to quickly and easily navigate their way throughout the GEnie service.

GEnie users now can effortlessly point and click on the next content area of their choice. In addition, an improved E-mail interface allows for direct connections across multiple online services with the simple click of an icon- - leaving behind confusing symbols or notations that can otherwise impede easy communication.

New Pricing Structure

GEnie will now offer an immediate price change that reduces prime time surcharges by 75% to $2 per hour, with anticipation of further pricing streamlining in the near future.

Full Internet Access

Effective April 11, GEnie now offers full, text-based Internet access, enabling users to tap into this global web of wide-ranging information and discussion groups. Specially designed Internet "launch pads," available only on GEnie, allow users to leap-frog to areas of particular interest with ease.

For novices, GEnie has also developed an Internet Education Center that dispenses helpful advice on getting around the global network of networks easily accessed with GEnie's new graphical interface.

Operational in 1985, GEnie Services is a pioneer in the online market, and a leading source of high quality, comprehensive electronic information and entertainment products. With a subscriber base that spans the world, GEnie is dedicated to providing value by meeting the exploding information and communications needs of businesses and consumers.

Headquartered in Rockville, Maryland, GEnie is part of GE Information Services, Inc.

New GEnie Rates

GEnie is happy to announce effective August 1, 1995, GEnie's 9600 baud and SprintNet surcharges will be eliminated. Along with the elimination of

high speed surcharges for both U.S. and Canada, we'll also offer 14.4 baud access through SprintNet and reduce Canada's Datapac surcharge to CAN $5.00/hour.

More GEnie news

November 13, 1995

Dear GEnie Subscriber,

GE Information Services, the owner of GEnie Online Services, announced yesterday that the company is working with the investment firm of Allen & Company to identify potential buyers for GEnie.

The decision to seek a buyer for GEnie was a difficult one for GE Information Services. We feel tremendous loyalty to our customers, and we considered very carefully the options available to us. Finally, our business decision was based on a desire to focus on GE Information Services' main mission -- to provide business productivity solutions to businesses around the world. We lead this market, and we derive more than 95% of our revenues from this market. In 1995, GE Information Services has invested in GEnie with a new graphical user interface, new multi-player game offerings, faster access speeds, and simplified pricing. We believe that GEnie presents a good fit within a company whose main focus is in the consumer market.

GEnie subscribers are the best in the world, and we are keeping you front and center as we seek a new owner for GEnie. We assure you that you will receive top quality service and support throughout this process. We will make every effort to make this transition transparent to you. In the long run, we hope that this transaction will make GEnie a better and even more exciting service for you, our customer.

Please check GEnie's logon announcements periodically. Updates will be provided whenever new information can be released. (Type "GENIE" at any menu prompt and select Item #2 from the menu which is presented to access the announcements if you're already online, or take a moment to review them each time you logon.) Sincerely, Horace Martin ,Acting President, GEnie Online Services

Babylon 5

Did you know Babylon 5, the show, is located at Grid Epsilon 470/19/5. That is GEnie's page 470, category 19, topic 5 in the Science Fiction RoundTable!

GEOS

Geoworks is pleased to announce the appointment of DPI Services as the exclusive republisher of Geoworks' English-language desktop product line, including the award-winning Ensemble package. DPI is an experienced software publisher and value added reselle

DPI assumed the manufacturing, sales and marketing and distribution responsibilities for Geoworks' line of English-language desktop products on March 1, 1995. The product line includes Ensemble, Quick Start, GeoPublish, Font Packs, Clip Art Libraries, Esc

For more information about DPI, contact them at 151 Martinvale Lane, San Jose, CA 95119, telephone (408) 629 3700, fax (408) 629 0141. To order products, call (800) 824-4558.

We are pleased to work with DPI Services and are confident that they will maintain the high level of product quality, service and support that our customers have grown to expect for Geoworks products

Questions and Answers:

Q. As a Geoworks customer, how will this change affect me?

A. For the most part, the transition has already taken place and will be transparent. A few phone numbers will change in April and we will announce those shortly.

Q. What are Quintessence Game Pack, Crossword, and Quick Designer Templates? How can I get them?

A. Crossword and Quick Designer Templates are new products developed recently by Geoworks for Ensemble users. Quintessence Game Pack was developed by LesInk Productions in cooperation with Geoworks. DPI will be introducing these new products shortly. Stay

Q. What about GEOS development tools like the SDK and Bindery?

A. Geoworks will continue to support its developer program and software development tools directly. For sales and product information about the SDK or Bindery, call (800) 436-7735.

Q. Why is Geoworks "out-sourcing" the promotion, sales and support of its desktop product line to DPI Services?

A. Quite frankly, Geoworks felt it was time to enroll the assistance of a company with proven experience in the marketing and support of desktop software. Both Geoworks and DPI Services are excited about their new partnership and will be working closely to make Ensemble more successful than ever.

PERFORMANCE PERIPHERALS, INC.
Product News Releases
MAY 1995

BBU - .Battery Back Up for REUs.

The BBU is a battery backup unit for 17xx series RAM expansion units, 1750 CLONE and GEORAM 512K. The BBU provides all power to the REU and therefore there is NO need for a heavy duty power supply for the computer. The batteries (not included) allow the BBU to continue to back up the REU when power to the wall mount supply is lost. A red BATTERY LOW indicator lights up when the voltage of the batteries drops from a fresh voltage of 6.0 volts to a used voltage of approximately 5.1 volts. If power to the wall mount power supply is never lost then the batteries will never be used by the BBU. When you feel an urge to see if your REU is being accessed then take a look at the green ACTIVITY indicator included.

The BBU is quite compact. It measures 2.5 inches wide by 4.5 inches long by 1inch tall. The REU fits vertically into the BBU. This adds some height to thecomputer but it shortens the depth on REU would normally take up on a desk.

Complete documentation for setup and use is included in the manual. And software for rebooting GEOS and 128 GEOS is included. The BBU works with many programs that allow reuse of data stored in and REU. RAMDOS and (RAMDOS II) work very well with the BBU. The RAMDOS programs allow the

use of the REU as a disk drive. There are many applications for the BBU and you have to try it to believe it. A Wall mount power supply, battery holder, manual and a utilities disk are included.

BBGRam

BBGRam - Battery Backed GEOS compatible RAM expansion unit. This battery backed non-volatile RAM expander is similar to the volatile GEORAM 512K from Berkeley Softworks. It is the RAM expander of choice for GEOS users. For GEOS users, the BBGRam is all the RAM expander that the user needs. The BBGRam has battery backup built in and comes in sizes of 512K, 1M and 2M. Compared to a floppy drive BBGRam wins on all counts. lower cost, much faster, no moving parts to break or wear out, smaller, and less power consumption. The 2M version stores the equivalent of ten 1541 disk drives. Additional features include a red BATTERY LOW indicator, and a green ACTIVITY indicator. Wall mount power supply, 4 D-cell battery holder, manual and utilities disk included.

BBRTC

BBRTC - Battery Backed Real Time Clock. This tiny module, plugs into the unused CONTROL PORT. The time and date is read out of the BBRTC automatically upon booting or rebooting GEOS. Additional software utilities are provided which allow the display of time and date from the BASIC command prompt. BASIC programs for controlling the BBRTC are included. A utilities disk and a user's manual are included.

RAMDrive

RAMDrive - The High Speed General Purpose Disk Drive Emulator. The Performance Peripherals' RAMDrive was developed together with the Creative Micro Designs' RAMLink. RAMLink and RAMDrive are very similar, and reach the same speed performance and level of compatibility with application software. RAMDrive is effectively a disk drive substitute. To software, RAMDrive looks and acts as if it were a disk drive; with one exception - it is very very fast. RAMDrive is also much faster than a hard drive and has no moving parts to wearout.. RAMDrive does work in combination with 17xx series REU on an expansion board. RAMDrive works with GEOS in addition to regular Commodore DOS programs. RAMDrive is very portable and has built in rechargeable batteries. The batteries last up to 2 weeks on a full charge and are recharged continuously from the power supply and built in recharge circuit. The RAMDrive manual is very precise and is an excellent reference for the extensive and complete features the RAMDrive offers. The RAMDrive provides auto-boot program execution on power up for both the C64 and the C128. The RAMDrive

is an extremely powerful and flexible product. Additional hardware features of the RAMDrive include ENABLE/DISABLE switch, (drive) SWAP8, SWAP9, RESET switch, and BBG (battery backed good) indicator, ACTIVITY indicator, and ERROR indicator. User manual and utilities disk included. Please note that for users interested in speeding up GEOS only, then the BBGRam is a more cost effective solution than the RAMDrive.

FLASH8

FLASH8 - 8 Megahertz Accelerator Card for the C64. 8 MHz accelerator imported from Germany. Works on a C64 only, not with a C128. Compatible with GEOS, and BASIC programs. GEOS at 8 MHz it truly amazing. Another highly successful application for the FLASH8 is BBS systems. Manual and utilities disk included.

64NET (beta version)

64NET authored by Paul Gardner-Stephens of Australia, allows the C64 to use the resources of an IBM-PC. A parallel cable from the C64 user port to the PC printer port connects the two systems. Up to four C64s can share a common PC and the PC's hard drives, CD-ROMs and other resources. Sample program files are included. This software is in beta testing; which means that the user may find bugs and undocumented operations. The beta version is recommended for advanced computer users only. When 64NET is released as a commercial package the version will be 2.xx. So if this is interesting to you, write to us so we can send you information when 64NET 2.0 is available. Various system configurations are available from PPI for this shareware product. A shareware demo version is available which is limited to read only from the PC. Cables, shareware and key-files are available. The key file allows use of all the features of 64NET.

geoCOM

Imported from Germany. geoCOM is a full featured GEOS programming system. The many features of this system allow application creation similar to the ease with which BASIC programs are created. Please write for additional information. English manual and disk included.

LOW RISK TRIAL PERIOD

A limited 30 day money back guarantee is available on the BBU, BBGRam, BBRTC, and RAMDrive. Products returned for a refund must be undamaged. Shipping and handling charges are not refundable. PPI reserves the right to charge a 15% restocking fee. Please call for additional details.

BBRTC	$ 24.97
BBU	$ 61.97
RAMDrive 1M	CALL
RAMDrive 2M	CALL
BBGRam 512K	$ 92.97
BBGRam 1M	$ 123.97
BBGRam 2M	$ 165.97
FLASH8	CALL
64NET beta key disk	CALL
64NET Cable	$ 19.00
64NET Shareware disks	$ 5.00
geoCOM	CALL

SHIPPING and HANDLING:

For 64NET Cable, FLASH8, RAMDrive, BBGRam and BBU, add: $6.00 for U.S., $10.00 for Canada, and $19.00 for other destinations.

For BBRTC add: $3.00 for U.S., $5.00 for Canada, $10 other. Cash on delivery (COD) for U.S. only: add $5.00.

All prices in U.S. funds only. Prices and terms subject to change without notice. Dealer pricing available.

ORDERS only:800-EASYWEB (800-327-9932).
Technical help Voice/FAX:518-436-0485.
Internet E-Mail:<p.fiset@genie.geis.com>.
TCCUG BBS:518-370-8632, handle: <periph>,
TCCUG BBS:SUB4 is Performance Peripherals' area.

Performance Peripherals, Inc. - 5 Upper Loudon Road
Loudonville, NY 12211 - U.S.A.

Networking

Dr.Octal's Q&A Clinic

By John W. Brown (JBEE)

Q: Could you recommend a good disk cataloguing program?
Mark Voorthuyzen

A: As for a good disk cataloging program, I can not recommend any because any that I have found do not allow the exporting and importing of data text files from within the program. Which means after you have spent your time logging in disks and want to share it with others you can not, unless you force others to use the same disk cataloguing program that you do. That is why I use Paperclip III to pull our public domain disk directories into text documents and then save them to disk as sequential text files. Everyone can read SEQ disk files and by using Paperclip III's sorting functions you can compare existing program directories against potential programs before adding the programs to your current disk collection. With Paperclip III you can add disk numbers after the program name and even comments, like we do. It may take a bit more work than a "dedicated" disk cataloguing program, but it is much more flexible and can produce nicer looking printouts too.

Q: How come on the first page of my catalog, the date is different from the date on the last page of my catalog (the order sheet). Various Customers

A: Because we print so many catalogs with so many pages it takes a few days sometimes to finish the mailing. The cover is usually printed out last. Also, the date is automatically inserted by our Desk Top Publishing (DTP) program when we are printing copies of the catalog.

Q: I have a C-64C, 1541-II, and a new 1581 drive. Whenever I load 1581 Toolkit or RUN Shell by RUN Magazine, my system will "sometimes" recognize the 1581 (#9). How can I get my computer to always recognize my 1581 drive?

I have disconnected all peripherals and changed device numbers to no avail. I have separated overlapping wires from behind my computer station and moved the drives away from the monitor. Can you help? - Charles J. Puccio

A: Assuming your disk drives are hardwired as device #8 and device #9, vs. using a software program to set the device number, the most likely problems are:

1) The programs use disk routines that only work properly on 1541s because they are seeking 1541 information in specific track/sector, ROM, or RAM locations within the disk drive.

2) The programs only work properly with device #8.

3) The programs use a speed loader which will freeze on a 1581 due to such things as timing problems.

4) You might have other hardware on your system, that is causing a serial bus timing problem. This is usually something such as a turned off device, a printer interface that uses the cassette port, or a "clone" printer or disk drive that was not designed properly that is causing the serial bus problems. You might try switching the drives so the 1581 is the first drive in line on the serial bus chain.

Since I did not have the software you use, I posted your question on GEnie and we received these replies:

From H.HERMAN1:

JBEE, I'd vote for your suggestions:

1) The software is not addressing the 1581 drive number, or

2) Try putting the 1581 as the first drive (no matter what its number) after the C64. I have the 1581 Toolkit but found it difficult to use, and was happy when I switched to using JiffyDos, instead. RUN Shell should run okey. No idea why this is a problem for the user. Howie

From CMD-DOUG :

I would suspect the user has a loading problem, a bad serial cable, or a flakey 1581. The program is certainly designed to handle the 1581 and device 9, so those items shouldn't be an issue.

Q: Where can I buy 80 column RGBI monitors for my C-128 (various)

A: Your best bet at this point is to buy a used RGBI PC monitor, sometimes called a "CGA" monitor from a place that sells used or refurbished computer

equipment. There are advertisers in Processor Magazine and Computer Shopper that sell them. I recently picked up a Nanao Flexscan 8060S "High Resolution RGB" color monitor. It work both in analog (PC) and RGBI (C-128) modes. The great thing is being a PC multi-scan monitor, not only is it far sharper then a original Commodore monitor but when I use an interlaced program such as Lace II I do not have to adjust the horizontal switches. I can highly recommend this particular model. I "heard" the NEC-2A multi-scan RGBI models will work nicely too. Though I have not personally tested them.

Subscription Information

The magazine is published 2-3 times a year.

For the US, Canada, Mexico, & PR the subscription cost with the companion disk, mailed by first class mail,
for issues 33-38 (six issues), is only $24.00
(Order item#901).

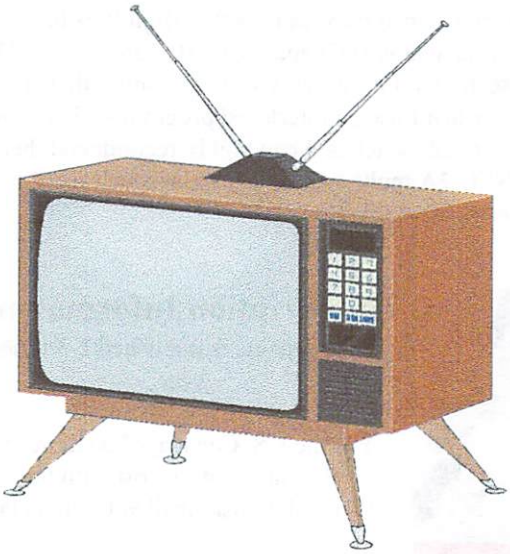
For all other countries including overseas the subscription with the disk, mailed by surface printed matter, for issues 33-38 (six issues) is $36.00 (Order item#902).The available single issues are #28-#36, item numbers #803-811, which cost from $2.00 to $6.00 each, plus S&H. Mail your orders to:

Parsec, Inc. POB 111 SALEM, MA 01970-0111 USA

Author Guidelines

1) It has to be good,
2) interesting,
3) previously unpublished,
4) or unique.
5) We must be able to easily move the graphics to GEOS.
6) We must be able to easily move the text to sequential files.
7) Authors have to be active GEnie members.

Leave me e-mail about what you have in mind or what you are interested in doing. I have tons of pet projects I will never get around to polishing off.



Display Alot

By John W. Brown

Display Alot
the video / lottery display program

This program is for the C-128 and uses both the 40 and 80 column modes of the C-128 (flat or D models) at the same time. It was written to display lottery numbers for a video store, or any store, and to display large letters on the 40 column screen while using the 80 column screen for text input.

This allows an uninterrupted display using the 40 column output which the customer can view on either a large TV screen or on a 40 column monitor, even while you are inputting data using the 80 column monitor. The 80 column monitor can be monochrome or color.

This program features:

- Automatic rotating display between any of the choices enabled.
- Display of the time (entering the time for the program is mandatory to prevent the program from displaying a blank screen to viewers).
- The display of a 4 digit lottery number from the night before.
- The display of a 4 digit lottery number from tonight.
- The display of a 5 digit lottery number.
- The display of a 6 digit lottery number.
- The display of a 7 digit lottery number.
- You can change the default colors (B&W is the default) using the menu or by changing the BASIC program values.

The unique features:

- Uses both the 40 and 80 column displays at the same time.

- Uses the C-128's powerful BASIC 7.0 graphic commands (GSHAPE etc.),
- It is written in BASIC 7.0 so you can easily modify it, compile it, or add the SID routines published in the SIDPLAYER book by COMPUTE, which Parsec distributes.
- Display Alot uses standard Doodle bit maps so changing font sets is easy.
- If you need programming and technical support it is provided on GEnie in the Commodore RoundTable (M625;1) in Category #11.

You can input "hot" messages entering up to a 10 character by 5 line message. You can display two default screens embedded within the program. You can also change the default 3 letter logo to one of your own by modifying the BASIC program. All letters must be entered in lower case, which is the default. The text shown on the 40 column screen is output in CAPITAL letters for maximum effect.

The input/output of data/letters supports the letters A-Z, the minus sign, the colon, and the period for output on the 40column screen.

While the program is running hold down any of the menu keys to stop the program, while it still displays a picture on the 40 column screen, to change any of the selections, including the time. The program will respond within a few seconds once it has finished updating the 40 column screen.

DisplayAlot features a text auto centering routine in BASIC so lines with only a few characters are displayed properly centered. This prevents the screen from looking lop sided. The characters are in a 32x40 matrix so the letters are clearly visible many feet away using an inexpensive 19 inch TV. The "hot keyed" special screen supports a 10 character by 5 line display for custom text display rotated among the enabled default screens, this feature was designed for the "special" of the day announcements or late breaking news.

What is needed for this program:

C-128 (flat or D model)

A C-1541 disk drive or any type of drive and device #.

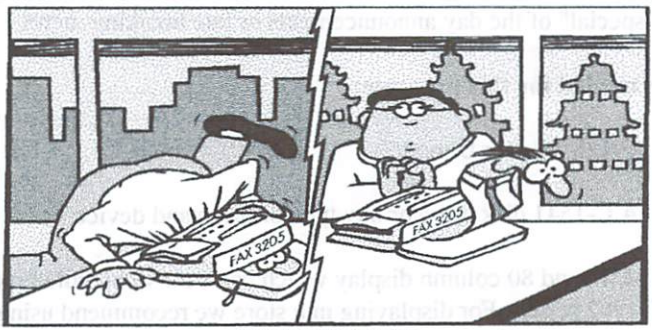
A 40 and 80 column display which can be a 40/80 column monitor such as the C-1902 series. For displaying in a store we recommend using a TV for the 40 column output and an inexpensive monochrome 80 column monitor for the input.

To modify this program you need an intermediate knowledge of BASIC programming, at the most. I structured the program with many GOSUBs and included a lot of REM statements to make the program easier to follow.

For any type of retail store, this program is much more cost effective then buying a PC scan converter (VGA->NSTC $300) and then tying up an expensive PC. It is also much cheaper than an LCD display unit ($350 and up for a 2x27 inch display unit) and much more flexible . A typical used C-128 system (C-128D $100, 1581 $50, monochrome monitor $50) can be bought for $200 or less.

Please note that the font files, which are Doodle format pictures grabbed by the program, once displayed are put into "strings" for the bitmapped 40 column screen font. You can change the font by changing the Doodle pictures. Though to save disk space I shorten the Doodle files so they might not work well in a drawing program expecting a true Doodle file. In that case, load the file in as a uncompressed bitmap.

Although the original version was all BASIC 7.0, which was my goal, for the 40 column clear screen routine I decided to use a little machine language since it was an instant clear vs. very slow in BASIC 7.0. I have included both versions on disk, the all BASIC 7.0 version and the version using BASIC7.0 with the small piece of machine language. Source code for the machine language is included. You can compile the BASIC 7.0 program if you wish. If you wanted to get creative, you could add a display routine to show Doodle format pictures, such as Christmas or holiday ones, between the 40 column text screens. Enjoy this unique program and have fun. See the files on side A of the companion disk: display1.sfx, display2.sfx, display3.sfx. Load and run all three files, in order, onto a disk with 664 CBM BLKS FREE.



FAX-FAUX

Legal Stuff

Twin Cities 128/64, the magazine, or companion disk, may NOT be copied in whole or in part for ANY reason. TC-128/64's companion disk is commercial software and is only for individual use. It may NOT be put into a User Group's "commercial disk collection" where it can be copied freely, loaned, or rented. If it is, the group and officers will be held liable for all attorney fees and damages.

Please note that the Public Domain and/or shareware disk that comes on the back side of the companion disk CAN be freely distributed.

SOFTWARE NOTICE, RIGHT TO USE

The software, hardware, and routines published in this magazine can be used free of charge only if ALL of the following conditions are meet:

1)The program is copyrighted but freely distributable and you were a subscriber when the issue was first published.

2)You have to give a written notice on your first screen or title screen, where this type of phrase can be clearly noted by the user (Ex:) "Sound Routines from Twin Cities 128/64 - issue #32".

3)You have to send us a copy of the program on a disk or upload the program to our library on GEnie. Notify "JBEE" when you do. Do not send it by e-mail!

4)If there are any kind of charges for the program, either as commercial or shareware software, or if it is a "demo" for a company, contact us FIRST before releasing the software/hardware so we can talk about the licensing fee. This usually will be something small, such as copy of the finished product. If we find out after the fact it will cost you *MUCH* more. Only written releases from Parsec through the U.S. mail with our company stamp imprinted on the contract will be considered valid.

5)These routines may not be uploaded to any network.

6)These routines may NOT put into ANY disk library collection - individual use only - no exceptions!

Dr.Octal's *Sharp Operating Tips*

<tm>

Tip #&0050
Subject: Database fields
From: Noel Nyman

When you use a database program for name/address records, you usually have options such as "character" and "numeric" for the individual fields in the record. Since the ZIP code is a number, you may be tempted to make ZIP a "numeric" field. DON'T DO IT! "Numeric" fields are used to store numbers for calculations. The computer will treat the data you enter as a value rather than a sequence of characters. In a "numeric" field these zips

02139 60611-1234

will be changed to

2139 59377

Computers do not print leading zeroes in numbers as a rule. Zip codes for Puerto Rico through New Jersey would all be four digit numbers with "numeric" zips. In the second example, the Chicago zip moved to Montana because the "-" used by the Postal Service to separate the codes in "ZIP+4" is interpreted as a minus sign in a "numeric" field. The computer does just what it is told to do: it subtracts 1234 from 60611.

Tip #&0051
Subject: A printer tip
From: Noel Nyman

To print right at the top or bottom of a single sheet, such as with Print Shop, tape two sheets of paper together. Use pink hair setting tape. It holds firmly, but peels away without making a nasty mess.

Tip #&0052
Subject: A printer tip
From: Noel Nyman

To get double spaced program listings, handy for making comments or corrections on the hard copy, use:

OPEN 128,4: CMD 128: LIST

If a file number between 128 and 255 is used in an OPEN statement, an extra <RETURN> is sent after each line, creating a double spaced listing. End the printer listing mode with

PRINT#128: CLOSE 128

Tip #&0053
Subject: A printer tip
From: Noel Nyman

You can send "control codes" (CHR$ numbers below 32) to a printer or other device by holding the <CTRL> key and pressing other keys on the keyboard. A-Z give the numbers 1-26...<CTRL-M> sends a CHR$(13), for example. The <CTRL-:) combination sends a chr$(27), the commonly used "escape" code. If your printer uses the sequence

<escape> 69

to switch to emphasized printing, you can send the code this way:

OPEN 4,4: PRINT#4, <CTRL-:> CHR$(69)

Tip #&0054
Subject: Using GEOS directories in word processors
From: John W. Brown (JBEE)

Assuming that the disk you want to read is in drive #9 (source) and the write (destination) disk is in drive #8. Type from Basic:

```
load "$",9
open8,8,8,"geosdir,s,w":cmd8:list:print#8:close8
```

Then read the sequential file "geosdir" into your word processor as an ASCII text file or convert it to a PET ASCII sequential file if you have to. This works in both C-64 or C-128 mode.

CLASS(Y) ads

CLASS(Y) ads can be submitted on either 1541 or 1581 disks as either PetAscii or straight ASCII sequential text files, printed by hand, or sent as hard copy from a printer. Parsec is not responsible for omissions or typographical errors. Ads will not be returned. BUYER & SELLER BEWARE! The guidelines to buying through the mail are unless you know the person well: **1)** Buyers and sellers should insist on COD, ship by UPS (if possible), cash or money order! **2)** Get a telephone number! **3)** Try to have some fun horse trading! **RATES and RULES** $2.00 a line, each line is 75 characters long, a minimum charge of $10 per ad. Only alphanumerical characters, allowed plus punctuation, no special characters, abbr. are okay. The Seller must be a TC-128/64 subscriber. Commercial advertisers have to submit a copy of their product for review. Your ad does not have to be computer related. We reserve the right to cancel any ad for any reason. Canceled ads will have their money returned. If there is not a category already created for your Class(y) Ad, then we will create one.

FOR SALE HARDWARE
FOR SALE SOFTWARE
COMMODORE DEALERS AND REPAIRS

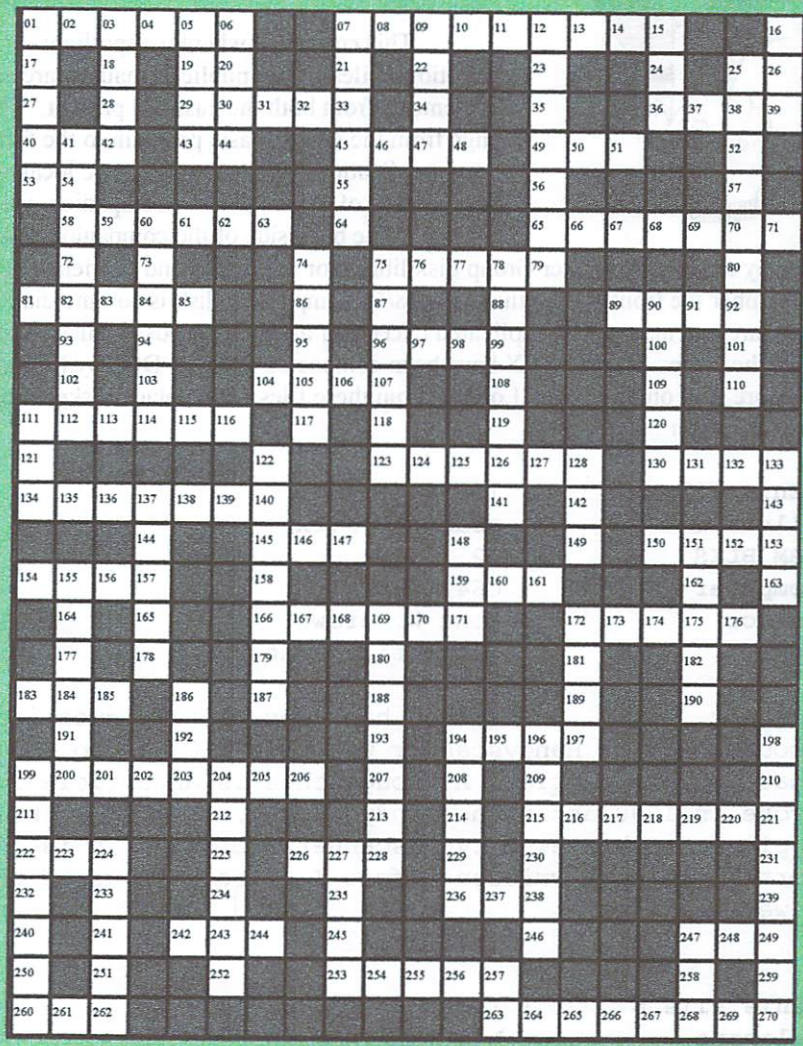
COMMODORE USER GROUPS

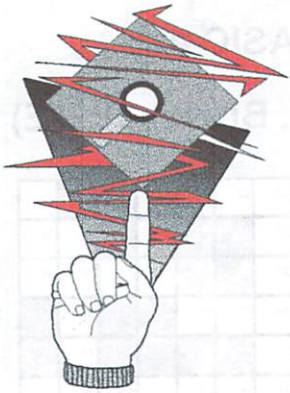
TPUG
Toronto pet Users Group Inc.
5334 Yonge St Box #116
Willowdale ONT M2N 6M2
Canada
416-253-9637

Large PD collection for all Commodore and PC computers.
Membership prices
Canada $25
International $30 US
USA $25 US

Puzzle: Something BASIC #1

CompuCross (tm) By John W. Brown (JBEE)





Software Picks

By John W. Brown (JBEE)

This column is where we spotlight exceptional files, either public domain, shareware, or "demos", from both the past and present. They range from the simple basic program to the full blown application. These programs are located on the back side of the TC-128/64 companion disk. It is (ONLY) the back side of the companion disk that is okay to copy for a User Group disk library or to pass around to friends. Please remember the front side of the TC-128/64 companion disk is commercial software and may not be duplicated except for archival purposes. Files ending with the suffix .SDA or .SFX have been compacted with CS-DOS by Parsec to fit more files onto the disk. Load and run these files onto a blank diskette to dissolve them.

GEnie file # : 12198
Filename : scanner-jb.txt
CBM BLKS : 42
Computer : C64/128
Author : John W. Brown
Type : sequential text

Comment : Parsec has been getting questions about what the Handyscanner was that RIO use to sell that CMD now carries. Although this is an article I wrote in 1990 about the Handyscanner, most of it still applies to the newer type scanner. For what it is worth, this was written *before* I edited and wrote the newer manuals.

GEnie file # : 17233
Filename : bug machine.sda
CBM BLKS : 95
Computer : C-64
Author : "the spy"?
Type : demo

Comment : A nice digi-sound treat. Compacted with CS-DOS by Parsec. Load and run onto a blank diskette to dissoolve.

GEnie file # : 17672
Filename : technitra-triad
CBM BLKS : 201
Computer : C-64
Author : triad
Type : demo

Comment : Another nice digi-sound demo. Sort of like the demo Digital Acid (the one with the bouncing smiley faces).

GEnie file # : 16809
Filename : guilty.arcan.sfx
CBM BLKS : 74
Computer : C-64
Author : arcane
Type : demo

Comment : This has a tie-in with the O.J. Simpson trail. Everything else in 1995 had one, so why not a demo? Compacted with CS-DOS by Parsec. Load and run onto a blank diskette to dissoolve.

GEnie file # : 17668
Filename : hummimg bird
CBM BLKS : 39
Computer : C-64
Author : unknown
Type : demo

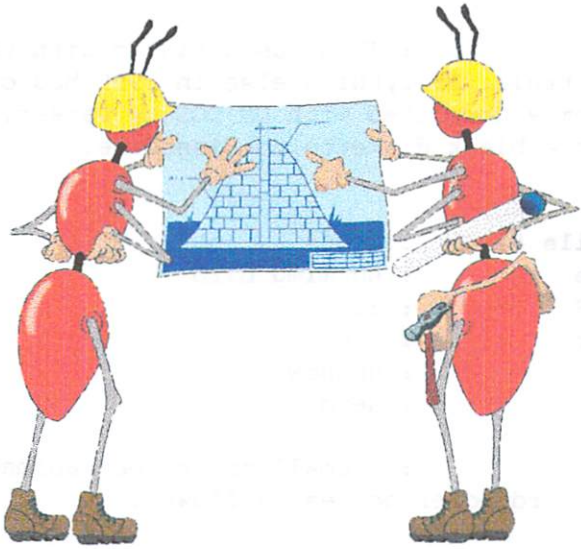
Comment : A small one screen animation of a humming bird hovering near a flower.

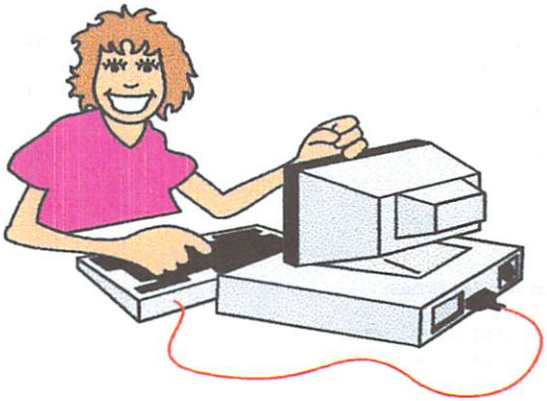
GEnie file # : 17625
Filename : b-station.sda
CBM BLKS : 126
Computer : C-64
Author : J.Green
Type : arcade game

Comment : An arcade type game where you defend your ship by blasting planes. Get the edge by taking advantage of the radar on the left to track the incoming enemies.

GEnie file # : 17710
Filename : strange xmas.sda
CBM BLKS : 53
Computer : C-64
Author : J.Green
Type : demo

Comment : A cute little story told with character animation.





Structured BASIC 64

V10.2

By JBEE

On 930807 Parsec, Inc, bought the complete rights to Structured BASIC 9.4 from Kent Smotherman. We have recently upgraded the package and slightly renamed the product. Anyone who that has bought the software from Parsec or a Parsec authorized dealer from 930807 to 951230 is eligible for a partial or full refund.

If you bought SB9.4 from 930807 through 951230 and are a TC-128/64 subscriber you are due a 100% refund of your money even if you do not normally get the companion disk. Just send Parsec a copy of your canceled check, front and back, to our POB.

If you bought SB9.4 from 930807 through 951230 and are NOT a TC-128/64 subscriber you are due a refund between the current selling price of $12 and whatever you paid. Just send Parsec a copy of your canceled check, front and back, to our POB.

If you bought SB9.4 before 930807 and are NOT a TC-128/64 subscriber you can update your older current version to the current version for only $5. That covers the shipping and handling plus registers you as a user. Just send Parsec a copy of your canceled check, front and back, along with the old manual and disk to our POB.

A mini-Table of Contents:

Article continued in Issue #38!

Quick Reference Guide

Structured BASIC 64
LICENSE AGREEMENT

NOTICE TO USER: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT.

1. LICENSE GRANT - Parsec, Inc. grants to you, as an individual, a non-exclusive right to use one copy of the SOFTWARE associated with this license for use on your computer. This license to use the SOFTWARE is conditioned upon your compliance with the terms of this Agreement. You agree you will only copy the SOFTWARE as necessary to use it in accordance with this license. You shall not use, copy, rent, lease, sell, modify, decompile, disassemble, reverse engineer, or transfer the licensed program except as provided in this agreement. Any such unauthorized use shall result in immediate and automatic termination of this license. This license is valid only for the original purchaser and may not be transferred. All rights not expressly granted here are reserved to Parsec Inc.

2. COPYRIGHT - The SOFTWARE is protected by United States copyright law and international treaty provisions. You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. You further acknowledge that title and full ownership rights to the SOFTWARE will remain the exclusive property of Parsec, Inc. and you will not acquire any rights to the SOFTWARE except as expressly set forth in this license.

3. LIMITED WARRANTY - Parsec, Inc. warrants that the SOFTWARE will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of purchase. Any implied warranties relating to the SOFTWARE are limited to ninety (90) days.

4. PARSEC,INC. DOES NOT WARRANT THAT THE SOFTWARE IS ERROR FREE. PARSEC, INC. DISCLAIMS ALL OTHER WARRANTIES WITH RESPECT TO THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY MAY LAST, OR THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

5. NO LIABILITY FOR CONSEQUENTIAL DAMAGES - IN NO EVENT SHALL PARSEC, INC. OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE DELIVERY, PERFORMANCE OR USE OF THE SOFTWARE, EVEN IF PARSEC, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL PARSEC, INC'S LIABILITY FOR ANY CLAIM, WHETHER IN CONTRACT, TORT OR ANY OTHER THEORY OF LIABILITY, EXCEED THE LICENSE FEE PAID BY YOU, IF ANY.

6. This license may be modified at any time by Parsec Inc. without written notice. A copy of the modified agreement will be available for the cost of a SASE. All changes are binding on the purchaser.

7. When the original purchaser no longer uses the software, the software and materials must be destroyed or sent back to Parsec, Inc. It can not be resold.

8. ENTIRE AGREEMENT - This is the entire agreement between you and Parsec, Inc. which supersedes any prior agreement or understanding, whether written or oral, relating to the subject matter of this license.

Trademarks - C-64, C-128, C-128D, C-1541, C-1571, C-1581, C-1700, C-1764, C-1750, C-1351, C-1080, C-1902, and C-1084 are trademarks of Commodore Business Machines. CMD HD /RamLink are trademarks of Creative Micro Designs. GEOS is a trademark of Berkley Soft Works (GeoWorks).

Support

Support will be provided on the GEnie network inside the Commodore 64/128 RT and through the US mail. All letters received become the property of Parsec, Inc.

Foreword

"Structured BASIC 64" was the result of over five years of intensive development, usage, putting it on the back shelf, and then redevelopment. The ultimate goal was not to provide the standard graphics and music commands commonly found in most C-64 BASIC extensions. Instead it was to provide a thoroughly modern BASIC language and environment, for programming and using the C-64, that was as powerful and flexible as possible. To achieve that objective in order to extend the life of the C-64, to a new generation of users and programmers who are eager to have an inexpensive yet modern and powerful personal computer in their own homes, we introduce Structured BASIC 64. But what I really hope is that you find "Structured BASIC" as useful and enjoyable as I have.

Introduction

"Structured BASIC 64" is more than just an enhancement to the BASIC v2.0 inside the Commodore 64 computer. Instead it is a complete and advanced new operating system and programming language that transforms programming and using the venerable C-64 into an entirely new experience.

"Structured BASIC 64" introduces the powerful techniques of structured programming to C-64 BASIC programmers as well as providing significant new features to the way in which the C-64 operates, such as the capability for disk-resident virtual memory ("program paging") and a full implementation of windowing.

"Structured BASIC 64" also provides several features found only in the most sophisticated of today's programming languages such as C, PASCAL,$and ADA. These features include pointer variables, limited generics, and indirectly referenced procedure calls. So if you are a C-64 programmer that is looking for more power, flexibility, and ease in programming, or if you are just getting started in programming and want to learn it correctly, you have certainly discovered the answer in "Structured BASIC 64".

The purpose of this documentation is not to instruct in the techniques of structured programming but instead to describe how all of the "Structured BASIC 64" features are used. Check your local bookstore for instructional books on the topic of structured programming.

To load "Structured BASIC 64":

LOAD"SB64-jbee",8 RUN

After this program boots the "Structured BASIC 64" system, all BASIC memory is cleared. Once booted, "Structured BASIC 64" remains in effect until you turn off your 64.

"Structured BASIC 64" Overview

As a computer programming language the Commodore BASIC r2.0 inside the C-64 is relatively simple. The key to this simplicity is the "unstructured" nature of BASIC v2.0. An unstructured computer language is characterized by its lack of program execution control statements, relying primarily on statements like GOTO to alter the normally sequential execution of a program. However this simplicity is actually a burden for the programmer because large and complex programs containing many GOTOs are very difficult to understand and maintain. While there can be no argument that the GOTO is a powerful BASIC statement, it lacks refinement. A good portion of "Structured BASIC" deals with this issue of refining and 'structuring' the function of the GOTO. In fact, "Structured BASIC" eliminates the need for the GOTO itself (and the GOSUB) completely, replacing it with even more powerful statements and concepts. "Structured BASIC" gives the programmer much greater control in programming while at the same time introducing order and structure.

In addition to the structuring aspect of "Structured BASIC 64" there are a number of new string functions designed to make string handling much easier. Complete substring access is provided as well as string access to the 16k of 'hidden' RAM underneath the BASIC and KERNAL ROMs. Also, there are new

functions for using relative disk files in both a safe and convenient manner. Relative files are a very powerful method for storing disk based databases and other data but unfortunately they are inconvenient to use and understand.

"Structured BASIC 64" brings the usefulness of relative files within easy reach of BASIC programmers. The editing of program files is expanded by "Structured BASIC" to include the ability to delete, move, copy, and renumber blocks of program lines to speed program development and editing.

As powerful as all of these features already listed are "Structured BASIC 64" includes several even more powerful features. Advanced programming languages such as C, PASCAL, and ADA provide special variables called pointer variables that are used to indirectly access other variables. This feature is also supported by "Structured BASIC" not only for the standard purpose of pointer variables but also as a means of passing parameters to procedures.

"Structured BASIC 64" advances the use of pointer variables one step further to make them typeless, which provides the very powerful and unique ADA-like feature of generic variables. This is a very modern concept not available in any other BASIC for the C-64. Another enhancement is the ability to CALL a PROCedure indirectly through variables, even pointer variables. This is another highly advanced feature provided only by a few very modern languages that run on computers much more expensive than the C-64.

One of the most powerful features of "Structured BASIC 64" is the ability to access disk-based libraries of procedures at run time and to load them dynamically from disk for execution ("program paging"). This technique effectively increases the memory available for program code from the approximately 38k bytes in BASIC v2.0 up to the total space available on disk! That could be 170k for a 1541 up to 790k for a 1581 and beyond for hard disk users. This is an ability whose usefulness is limited only by the imagination, and one provided for your C-64 only by "Structured BASIC 64".

Perhaps the most stunning of "Structured BASIC's" features is windowing. Never before has the C-64 been provided with a feature quite like this one. Other programs may use windows but they do not allow the programmer to really use a true windowing system. "Structured BASIC 64" gives you windows not only for your programs to use but also for you to use from direct mode to display many types of data such as disk directories and program listings on the same screen, at the same time.

No modern programming language is complete without debugging and error trapping facilities. "Structured BASIC 64" offers a program trace feature that will remember up to the last 60 executed statements and display them on request. And "Structured BASIC 64"'s error trapping and handling is the most advanced of any language, allowing fully reentrant code.

Getting Started

"Structured BASIC 64" introduces the techniques of structured programming to the C-64, which to the unfamiliar may seem a very foreign way to program. The GOTO is shunned in structured programming as a primitive means for forcing program control. Instead "Structured BASIC 64" provides the concept of the program block. A program block is simply a section of program code that performs a specific task and whose beginning and end are marked by special BASIC instructions. BASIC v2.0 only has one true program block structure, the FOR/NEXT loop. Any other type of program block (or 'loop', a minor misnomer) in BASIC v2.0 is made with the GOTO, which can lead to clumsy and hard to read code. Consider this simple example of a small section of code to input and sum numbers until a zero is entered:

```
10 INPUT "NUMBER";N
20 S=S+N
30 IF N<>0 THEN 10
40 PRINT "SUM IS"S
```

This small section of code is actually a program block, that is it performs a specific function, that of summing the input numbers. What is not obvious at first glance is where the block begins and ends. Remember that this small code section could be contained in a larger program. Without memory wasting REMs to indicate where it begins and ends the task of identifying this program block may not be easy. Also note that the location, line numbers, of this small code section determines the target GOTO line number in line 30. If this code section were to be appear elsewhere in the program then this target GOTO line number would also have to be changed. Imagine, if you have not yet experienced it, the awkwardness of maintaining a complex program strewn with such GOTO-driven program blocks! But with "Structured BASIC 64" you can do it this way:

```
10 DO
20 INPUT "NUMBER";N
30 S=S+N
40 LOOP UNTIL N=0
```

```
50 PRINT "SUM IS"S
```

Now our simple program block is easily recognized as it is bounded by the **DO** and **LOOP** statements. We will discuss these statements in detail in a later section but for now just note how much more clear the purpose and location of the program block is in "Structured BASIC 64". Also note that this program block could appear anywhere within the program with no modification necessary. This is because line numbers in "Structured BASIC 64" are used only to input the program lines, not to control the execution of the program.

The "Structured BASIC 64" concept of the program block is closely related to a very important and central idea in "Structured BASIC 64": The concept of what is termed "context sensitive", or more appropriately "nesting sensitive", program design. This can be a difficult idea to grasp but hopefully another simple DO/LOOP example can help clarify:

```
10 DO
20 :DO
30 : I=I+1:PRINT I+J*5;
40 :LOOP UNTIL I=5
50 :I=0:J=J+1
60 LOOP UNTIL J=5
```

Note the use of leading : and blanks to indent program blocks. This is a good habit to adopt as it adds immensely to program readability. This simple code section will print the numbers 1 through 25. The question is how does "Structured BASIC 64" know which DO the LOOP in line 40 'belongs' to? Just as in BASIC v2.0 FOR/NEXT loops, it belongs with the DO that was last encountered, which is the DO in line 20. This is called nesting, where the DO/LOOP program block from lines 20-40 is said to be nested within the DO/LOOP program block from lines 10-60. In "Structured BASIC 64" the ability, that you will see in later sections, of affecting the nesting execution of program blocks is a very powerful feature, even more so than the GOTO. The statements that you will learn that have this ability are termed "nesting sensitive", and this will be explained in greater detail where it applies.

Structured BASIC Encyclopedia

Each of the "Structured BASIC" statements and functions will be detailed in this section, organized functionally and alphabetically. Their formats are described and examples are given to illustrate usage.

Program Execution Control

CALL string

The CALL instruction functions exactly as the GOSUB in BASIC v2.0, except that a target line number is not specified. Instead a name is given to the subroutine (see PROC) and that name is specified by the CALL instruction. Ex: CALL "SORT" . Since no line number is needed to tell CALL where the routine "SORT" is located within the entire program, "SORT" must be defined anywhere within the program by the PROC instruction. CALL may also be followed by a string variable instead of a string literal. This capability is very powerful since it allows for a tremendous flexibility in modular program design.

PROC"name"

PROC defines the start of a subroutine called "name". There is one restriction on the use of PROC: only 60 subroutines can be defined. But if you need more than 60 subroutines you must be working on an incredible program! When CALL"name" is executed the statement following PROC"name" is executed next. IMPORTANT NOTE: The PROC statement must be on a program line by itself with no other statements. Execution continues in a subroutine until the RTRN statement is encountered.

RTRN

This statement performs the return from subroutine function that RETURN does for GOSUB. Program execution is continued at the statement following the CALL.

FINI"name"

This instruction marks the end of the PROC. It is really the standard "Structured BASIC" end of block statement, but more about that will be described later. In terms of the PROC instruction FINIsh simply tells "Structured BASIC" where the PROC is completely finished. This allows for multiple RTRN instructions for the same PROC. Example:

```
10 FOR X=1 TO 3:CALL"SMALL":NEXT
20 PROC"SMALL"
30 :IF X=1 THEN PRINT"ONE ";:RTRN
40 :IF X=2 THEN PRINT"TWO ";:RTRN
50 :PRINT"THREE ";:RTRN
60 FINI"SMALL"
70 PRINT"DONE"
```

When RUN this program will print:

ONE TWO THREE DONE

This points out another aspect to PROC: If a PROC is encountered without being CALLED then the entire PROC is simply skipped. This happened in this program when statement 10 was finished executing. Statements 20-60 were skipped and then statement 70 executed. This allows for PROCs to appear anywhere within a program. Also notice that the "name" is not mandatory on the FINI statement and is not checked to match the PROC.

SELECT"name"/CASE expression/OTHER

In the previous example lines 30-50 where IF statements that selected what to do based on the value of X. Let us rewrite the PROC"SMALL" to use the "Structured BASIC" CASE program block structure:

```
20 PROC"SMALL"
30 :SELECT
40 : CASE X=1:PRINT"ONE":FINI
50 : CASE X=2:PRINT"TWO":FINI
60 : OTHER:PRINT"THREE":FINI
70 :FINI
80 RTRN:FINI"SMALL"
```

Note again the standard optional "name" on the SELECT. SELECT will select the first CASE statement whose expression evaluates to true and executes all the statements and lines between that CASE and it's FINI. If none of the CASE expressions are true then the OTHER portion is executed, if present. Once the CASE or OTHER is done then execution skips to the instruction following the FINI for the SELECT.

IF...BEGIN...ELSE

The blocked IF is used when an ELSE block is desired or if multiple lines are to be executed as part of the THEN (BEGIN) block. The blocked IF looks like this:

IF expression:BEGIN...FINI:ELSE...FINI

If the expression is true then all the statements and lines between BEGIN and its FINI are executed. If the expression if false all statements and lines between the ELSE and its FINI are executed. Note that the ELSE...FINI block is optional and that there is no explicit THEN -- the BEGIN replaces THEN when signifying that this is a blocked IF and not a BASIC v2.0 IF. As is standard in "Structured BASIC 64", BEGIN, ELSE, and their FINIs may include optional "name"s. IMPORTANT NOTE: When using the BASIC v2.0 IF in your

programs, if the statement following the IF is a "Structured BASIC 64" statement you must precede it with a colon:
IF X=0 THEN:CALL"SORT"

DO"name" [UNTIL/WHILE expr]...LOOP"name" [UNTIL/WHILE expr]

As usual, "name" is optional. The DO statement opens the loop and LOOP marks the end. The optional UNTIL or WHILE performs a conditional exit from the DO/LOOP: UNTIL exits the loop if its expression is true, WHILE exits if it is false. All statements and lines between the DO and its LOOP are executed repeatedly unless an UNTIL or WHILE causes the loop to terminate. Example: 10 DO:GET A$:LOOP UNTIL A$<>"" An addition method exists to force a DO/LOOP to terminate, the EXIT statement.

EXIT"name"

This will immediately terminate either the last open program block encountered or the one with "name" defined, if "name" is specified. Terminating a block refers to transferring execution to the statement following the block's FINI or LOOP. If "name" is specified then all open nested blocks from the current to the "name"d one are closed and the "name"d block is then terminated. Consider this example:

```
10 DO"STUFF":X=X+1
20 :IF X=4:BEGIN"CHECK"
30 : PRINT"ALL DONE"
40 : EXIT
50 :FINI"CHECK":X=0
60 LOOP"STUFF"
```

RUNning this program will result in "ALL DONE" being printed until the RUNSTOP key is pressed. This is because the EXIT in line 40 does not exit the DO/LOOP, but instead the last block that was opened prior to the EXIT, which is the BEGIN block in line 20. But by changing the EXIT in line 40 to EXIT"STUFF" then the DO/LOOP, which is named "STUFF", will be terminated. Additionally this will close the BEGIN block since it is nested within the DO/LOOP. This is what is meant by "nesting sensitive" design. In this example, EXIT"STUFF" is "nesting sensitive" to the BEGIN block and knows to properly close it in the process of terminating the "STUFF" block specified by the EXIT.

AGAINF"name"/AGAINB"name"

These two statements are used inside a DO/LOOP block. AGAINF, short for AGAIN FORWARD, immediately transfers execution to the LOOP statement in the DO/LOOP block that contains the AGAINF. AGAINB, short

for AGAIN BACKWARD, transfers execution to the DO in the current DO/LOOP block. If the optional "name" is specified then execution is transferred at the appropriate DO/LOOP named and all nested blocks are automatically closed as with EXIT. IMPORTANT NOTE: When using "name"d AGAINF/B be sure that the referenced block contains the AGAINF/B. Specifically, a PROC should not attempt to issue an AGAINB/F or EXIT to a "name"d block outside the PROC.

Functions and Pseudo Variables

INITS(length,character)

This function returns a string with the given 'length' and filled with the specified 'character'. For example: A$=INIT$(5,"A") would assign A$ the value of "AAAAA".

REPLACE$(string,start[,length])

This pseudo variable (the function call appears on the left of the '=' not the right) replaces a portion of 'string' with another string beginning at the 'start' character position for 'length' or the length of the assignment string, whichever is shorter. For example: A$="ABCDE":REPLACE$(A$,3,2)="1234" yields a value of "AB12E" for A$.

INDEX(string2,string1)

This numeric function returns the location of 'string2' within 'string1'. If it is not found then INDEX returns a 0. For example: PRINT INDEX("AT","CATS") prints 2.

MEM(address,length)

This function call can be used on either the left (pseudo variable) or right side of the '='. On the left it places the assignment string into memory at 'address' for 'length' characters. If 'length' is shorter than the length of the assignment string then the assignment string is truncated. If 'length' is longer then zeros are placed following the string. On the right of '=' MEM returns a string from 'address' for 'length' characters, even if the address range includes the RAM under the ROMs ($A000-BFFF and $E000-FCFF, see section 8 for further restrictions). For example: MEM(60960,5)="ABCDE":PRINT MEM(60960,5) will print "ABCDE".

AT(column,row)

This function is used in a PRINT statement to position the cursor at 'column' (0-39) and 'row' (0-25). Example: PRINT AT(20,13);"HI"

IMPORTANT NOTE: A semicolon and at least one print item must follow the AT(). Also note that AT functions relative to the current display window.

File Access

INCLUDE"filename",device#

This direct mode statement appends the "filename" from 'device#' to the BASIC program currently in memory. A RENUM will probably be necessary afterwards to resequence the line numbers.

RECIN#filenum,length,string var

This statement performs RECord INput of 'length' characters from the already opened 'filenum' into 'string var'. Reading of characters is not stopped by returns, commas, semi-colons, or colons as with INPUT#. For example:

```
10 OPEN2,8,2,"$":RECIN#2,142,A$
20 RECIN#2,16,B$:PRINTB$:CLOSE2
```

When RUN the disk name in drive 8 will be displayed. If the file associated with 'filenum' is for the modem (device# 2) then RECIN will wait up to 10 seconds between received characters before a time-out occurs, in which case a null string is returned.

RELIN#filenum,device,name,recordnum,length,stringvar

This statement performs input from the relative file 'name' from 'recordnum' on 'device' for 'length' characters into 'stringvar'. RELIN# is equivalent to performing the following statements:

```
10 OPEN15,device,15:OPENfilenum,device,filenum,name
20
RINT#15,"P"+CHR$(fn)+CHR$(rnlow)+CHR$(rnhigh)+CHR$(1)
30 RECIN#filenum,length,stringvar
40 CLOSEfilenum:CLOSE15
```

Where fn is filenum, rnhigh is INT(recordnum/256), and rnlow is recordnum-256*rnhigh.

RELIN*filenum,recordnum,length,stringvar

This variant of RELIN performs exactly the same function as RELIN# except it does not open or close any files. RELIN* assumes the disk command channel is opened as file number 15 and that 'filenum' is already opened as a

relative file. RELIN* performs all the lines in the equivalent program listed for RELIN# except lines 10 and 40.

RELOUT#filenum,device,name,recordnum,stringexpr

This statement performs relative file output. The parameters function similar to RELIN# except 'length' is not specified, it is taken as the length of the 'stringexpr'. This length should match the record length of the relative file 'name' to ensure proper file integrity. See the section "Notes On Relative Files" for details. In the RELIN# equivalent program substitute PRINT#filenum,stringexpr for the RECIN# in line 30.

RELOUT*filenum,recordnum,stringexpr

As with RELIN*, this RELOUT# variant performs no file OPENs or CLOSEs but otherwise functions in the same manner as RELOUT#.

Editor

FIND text

This direct mode statement searches the program text area looking for 'text'. It displays the line numbers of all BASIC lines containing 'text'. Note that any keywords contained in 'text' are tokenized prior to the search (unless enclosed in quotes). Example: FIND PROC will find all PROC statements in the program in memory.

RENUM step,startline#,endline#,newline#

This direct mode command renumbers the BASIC program line numbers. Renumbering begins with the first line# greater or equal to 'startline#' and ends with the first line# greater than or equal to 'endline#. The renumbered lines will start with 'newline#' and will increment by 'step'. If no parameters follow RENUM then the entire program is renumbered with an increment of 1. If only 'step' is specified then the entire program is renumbered with an increment of 'step'. Note that any target GOTO or GOSUB line numbers are NOT renumbered.

DEL start,end

This direct mode statement deletes the range of lines from 'start' to 'end' inclusive.

MOVE start,end TO to

This direct mode statement deletes the block of lines between 'start' and 'end' inclusive and inserts them at 'to'. All moved lines have their line numbers set to zero so a RENUM is necessary after the MOVE.

COPY start,end TO to

This statement functions like MOVE but does not delete the range of lines.

Pointer Variables

"Structured BASIC" introduces a new means of accessing variables called pointer variables. A pointer variable does not have a meaningful value itself, but instead is used to access the value of another variable. For those unfamiliar with pointer variables an example may help clarify exactly how a pointer variable is used:

```
10 A=5:P=@A:PRINT [P]
```

When RUN this short program prints 5. The statement P=@A assigns the variable A to the pointer variable P. It does not assign the value of A directly, but instead the variable A itself. In order to access the value of a pointer variable enclose the pointer variable name in brackets as the example does in its PRINT in line 10. Try adding this line:

```
20 A=10:PRINT [P]
```

After RUNning 5 then 10 will be printed. Since P is a pointer variable to A, when A is assigned the value 10 in line 20 P inherits this value as well automatically. What is really happening is the statement P=@A in line 10 sets P to actually be A -- from that point on, until P is assigned a new value there is no difference between coding A and [P]. Now add this:

```
30 [P]=15:PRINT A
```

RUN and see 5 then 10 then 15 print. Since there is no difference between A and [P] either may be used interchangeably. A pointer variable itself really has no type -- although it appears as a real variable it can be assigned to point to variables of any type, even to arrays. This means that the same pointer variable can be used to point to different variables at different times. This feature of the "Structured BASIC 64" pointer variable has a very important function: PROCs can be written that do not depend on the type of the data they process. For example, a single PROC to sort arrays of real, integer, and string types can be coded:

```
10 PROC"SORT"
20 F=0:DO UNTIL F=-1:F=-1
```

```
30 FOR I=S TO E-1
40 IF [P1](I) > [P1](I+1):BEGIN
50 T=[P1](I)
60 [P1](I)=[P1](I+1):[P1](I+1)=T
70 F=0:FINI
80 NEXT:LOOP:RTRN:FINI"SORT"
90 DIM A%(20)
100 FOR I=0 TO 20:A%(I)=RND(0)*100:NEXT
110 S=0:E=20:P1=@A%:CALL"SORT"
120 FOR I=0 TO 20:PRINT A%(I);:NEXT
```

The parameters for this SORT routine are shown in line 110: S is the starting element number to sort, E is the ending element number, and the pointer variable P1 is set to point to the array to sort. By changing lines 100-120 for different array types like real or string the PROC"SORT" will still perform the sort without any internal modifications necessary.

Program Paging

"Structured BASIC 64" provides a somewhat unique feature called program paging. This feature allows subprograms to reside on disk (or tape) and only be PAGEd into memory when they are needed to execute. What this means is that libraries of subprograms can be developed and stored on disk for use by any main program without the need to take up valuable program code space in the main program. Applications can be written modularly with shared paged subprograms that extend the amount of code available for the application. This saves disk storage by not having to duplicate the same subprograms in every program that needs them and also shortens the length of the program in memory since some of its subprograms will be stored on disk as paged programs. Program paging is similar to the program chaining ability of the LOAD statement except that the main program is not overwritten by the PAGEd subprogram. In other respects a paged program is similar to a PROC except that there is no PROC statement and the RTRN is replaced with the QUIT to return execution to the paging (CALLing) program.

Additionally since the PAGEd program resides in its own separate area of memory with its own variables there is no problem with paged programs having variables of the same name as those in the calling program. Block structures are shared for nesting purposes however so PROCs defined in the calling program can be CALLed by the paged program. However the main

CompuCross Solutions

Across
001.window
007.collision
019.pi
025.do
029.enter
036.wait
040.let
043.no
045.copy
049.and
053.en
058.verify
065.collect
075.fetch
081.close
089.auto
096.list
104.fast
111.delete
123.record
130.load
134.monitor
145.and
150.quit
154.trap
159.sin
166.header
172.begin
183.tan
194.swap
199.renumber
215.pointer
222.sys
226.key
236.pen
242.run
247.vol
253.tempo
260.end
263.rspcolor

Down
001.while
003.next
005.open
006.wino
007.circle
009.loop
012.scratch
015.new
016.pot
025.directory
041.envelope
060.resume
074.bload
075.filter
078.catalog
090.until
111.dim
122.graphic
128.dec
133.data
137.input
148.usr
151.using
155.rename
169.dverify
172.bump, 186.sum, 194.sleep, 196.append, 198.sprcolor,199.restore, 204.backup, 224.sound, 227.exit, 247.val, 257.or

program cannot CALL a PROC defined in the PAGEd program and upon execution of the QUIT all PROCs defined in the PAGEd program are discarded. Several instructions support program paging:

RESERVE num

This statement reserves 'num' blocks (256 bytes) of memory for the paged programs. RESERVE should be executed before any files are opened or any variables assigned since it performs a CLR. This memory is reserved at the top of BASIC memory. In general a RESERVE is only needed once in a program since each RESERVE takes away memory from the current top of BASIC and all PAGEd programs will execute in the same reserved space. Just issue a RESERVE for the largest of the paged programs at the start of the main program. The only way to safely restore RESERVEd memory is with the REGAIN statement (described later).

LOCAL (expression)

LOCAL can be used as a statement, function, or pseudo variable. As a statement 'expression' is omitted and LOCAL will select the paged program's memory and variables as the default configuration. Any variables or GOTOs will be made with reference to the paged programs memory after a LOCAL is issued. When LOCAL is used as a function 'expression' takes the form of an expression in parenthesis that will be evaluated using the paged program's memory without effecting the current default configuration setting (global or local). As a pseudo variable 'expression' is a single variable name in parenthesis from the paged program's memory that will be assigned the value from the right of the '=' in the assignment statement evaluated with the memory of the default configuration.
See the examples at the end of this section for clarification.

GLOBAL (expression)

This statement is used exactly like LOCAL except is selects the main programs memory and variables in all usages.

(THIS ARTICLE IS CONTINUED IN ISSUE #38!)