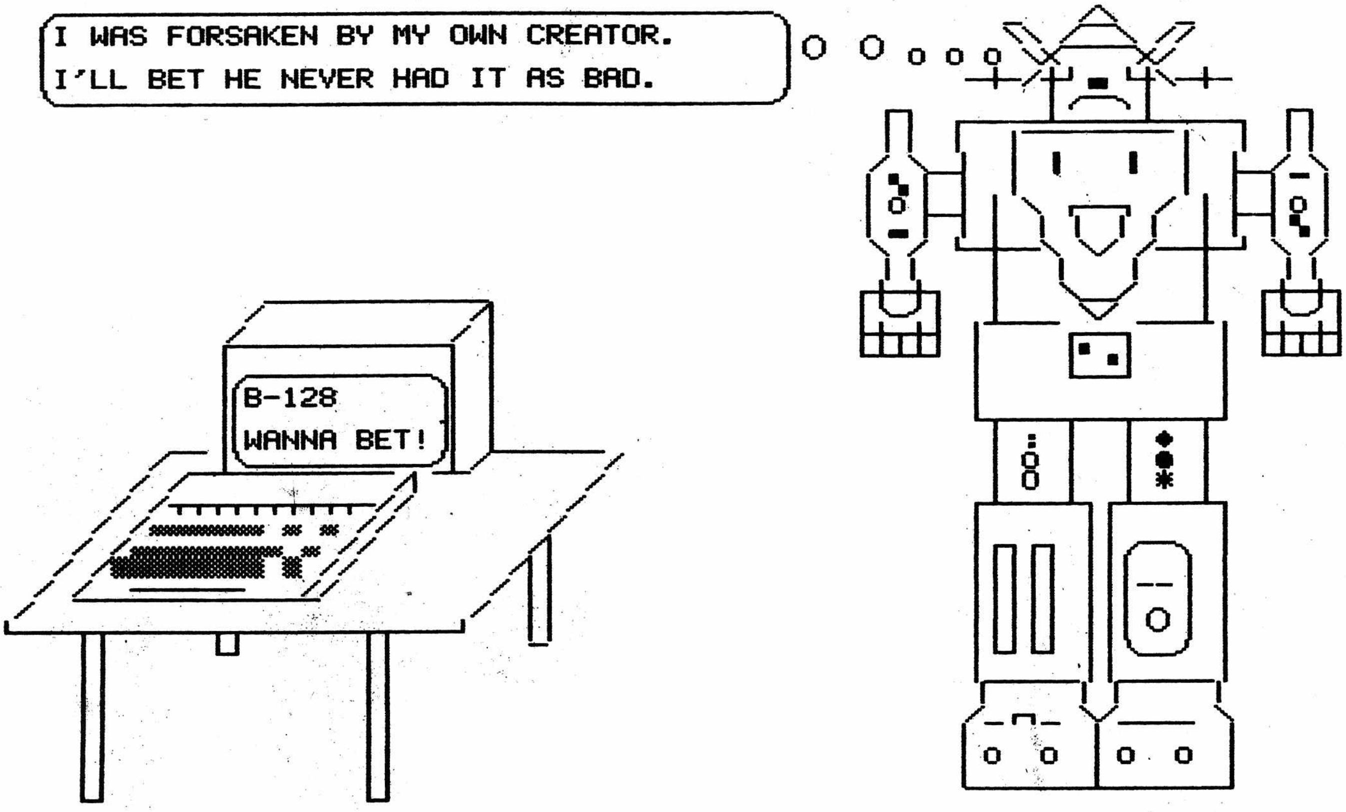
THE CBUG ESCAPE

CHICAGO B128 USER'S GROUP -- INTERNATIONAL
4102 N. Odell -- Norridge, Il. 60634 USA

- -

Return the ENTIRE Page -- including the mailing label. Make corrections as necessary
Please help us out. Print very legibly or type

UNEMPLOYMENT LINE


I WAS FORSAKEN BY MY OWN CREATOR.
I'LL BET HE NEVER HAD IT AS BAD.
B-128
WANNA BET!

TROY B.

From a disk by Troy Becker, program on Pre-Release #3

ORDER FORM -- USE PHOTOCOPIES FOR MULTIPLE ORDERS. EXTRA ORDER BLANKS WILL BE INCLUDED WITH SHIPMENT

NAME ______________________________

ADDRESS ______________________________

CITY ______________ STATE ______ ZIP __________

THIS IS YOUR SHIPPING LABEL -- PRINT OR TYPE NEATLY

ORDERS OVER 4 DISKS OR PHYSICAL EXAM SHIPPED BY BY UPS WITHIN CONTINENTAL U.S. YOU MUST USE STREET ADDRESSES FOR SUCH ORDERS -- UPS CAN NOT DELIVER TO POST OFFICE BOXES!

Your phone # ______________________________

Checks/Money Orders MUST be payable to NORMAN DELTZKE.
All payments must be in US funds. Send orders to CBUG c/o Norman Deltzke, 4102 N. Odell, Norridge, Il. 60634 U.S.A

Quantity	Stock #	Description	Price	Extension
________	10017	SSDD Premium Opus Disks/Pkg of 10 w/ sleeves & labels	$ 8.50	________
		Illinois residents add 7% Sales tax above item only		________
________	12881	CBUG #3 SWAN'S UTILITIES	14.00	________
________	12896	CBUG #4 KERNAGHAN'S UTILITY	9.00	________
________	12932	CBUG #7 SUPERBASE APPLICATIONS #1	9.00	________
________	12257	CBUG #11 CBUG TERMINAL PROGRRAM COLLECTION w/ B-Term	14.00	________
________	12261	CBUG #11a CBUG TERMINAL PROGRAM COLLECTION w/o B-Term	9.00	________
________	12984	CBUG #12 SCOTT'S B-MON	14.00	________
________	12787	CBUG #13 NORTHROP/LEIGHFIELD SUPERBASE AIDS	9.00	________
________	12684	CBUG #14 MEDICAL ACCOUNTING w/ Superbase	9.00	________
________	12716	CBUG #15 FRIENDFAM - Superbase	14.00	________
________	12773	CBUG #16 SWAN's BASIC 4.0+ COURSE	19.00	________
________	12670	CBUG #17 LIZ's UTILITIES, v1.2 (use back page order form for upgrade orders -- this number for new orders only)	16.00	________
________	12792	CBUG #18 GAMES & EDUCATION	10.00	________
________	12805	CBUG #19 THE WRITINGS OF MARLIN & FRIENDS	10.00	________
________	12768	CBUG #M20 UTILITIES & MISC #2	9.00	________
________	12699	CBUG #21 RETAIL NEWS DISTRIBUTION MANAGEMENT PROGRAMS	9.00	________
________	12701	CBUG #22 MATH EDUCATION PROGRAMS	9.00	________
________	12735	CBUG #23 BIBLE GAMES	15.00	________
________	12720	CBUG #24 THE 8432 EMULATOR DISASSEMBLED	9.00	________
________	12665	CBUG #25 PRINT FILES #3	9.00	________
________	12651	CBUG #26 PRINT FILES #2	9.00	________
________	12749	PRE RELEASE #4	9.00	________
________	_____	______________________________	_____	________
________	_____	______________________________	_____	________
________	11077	THE BUTCHERS HALF DOZEN FREE-BEE: Order any 6 disks from the CBUG library listed above, and get a 10 count bag of Premium OPUS SSDD disks absolutly free. CHECK HERE	FREE	________ □
________	12204	KNIGHT'S 8050 (DOS 2.7) COPY UTILITY	20.00	________
________	12276	BEELINE ADVANCED TERMINAL PROGRAM for Lemdata Software	35.00	________
________	12219	PHYSICAL EXAM 8050 -- same for all versions of 8050	35.00	________
________	12223	PHYSICAL EXAM 1541 -- For the 1541 drive	35.00	________
________	12238	PHYSICAL EXAM 4040 -- For the 4040 drive	35.00	________
________	12401	Recopy fee remittance	5.00	________
		BALANCE FORWARD FROM ARCHIVE/UPGRADE ORDER FORM (ATTACHED)		*________*
		MERCHANDISE SHIPPING AND HANDLING CHARGE --- ALWAYS INCLUDED, NO EXCEPTIONS		$ 2 . 0 0
		SUBTOTAL	$	________
________	12384	SUBSCRIPTION FEE Only paid up members may order! US, Mexico & Canada $ 10.00; Elsewhere $16.00 -- thru 12-86		________
________	12449	Extra copy of this publication (Winter/Spring 1986 ESCAPE)	6.00	________
________	12416	Additional free will contribution to CBUG		________

BY CHECK ___ BY MONEY ORDER ___ TOTAL REMITTED $ ______________

NOTE We endeavor to ship against money order orders within 3 business days of receipt. Orders paid paid by common check will be delayed 14 to 20 days.

THE CBUG ESCAPE

SECOND ISSUE
Winter/Spring 1986

* *

THE CBUG ESCAPE is a 4 times a year publication of the Chicago B128 Users Group International (CBUG), an international membership organization in support of applications and usage of the B128 Computer.

CBUG is NOT affiliated or allied with any other organization, users group, business or other entity of any kind, except in support of CBUG chapters.

Advertisements, articles and contents of disks are solely the responsibility of their authors. Their existence in a CBUG publication implies no endorsement by CBUG.

Publishing address: c/o Norman Deltzke, 4102 N. Odell, Norridge, Il. 60634 USA

Cover price: $6.00. $10.00/yr subscription (US, Canada & Mexico)
$16.00/yr elsewhere

* *

TABLE OF CONTENTS

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ PAID ADVERTISEMENTS ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

INCOME TAX PROGRAM FOR THE B128. Let the B128 do calculating and transfering to other forms for you. Prints out to computer paper for your records. Present program consists of 1040,A,B,C,F,SE and Form 4797, Price $65.00. These programs operate within Calc Result ONLY. Custom Program is $20.00 PLUS $5.00 for each page of forms (list needs and send forms not in above program). MO's are fast. Support by telephone, paid calls only.
JIM O'HALLORAN, RT.2 OWL CREEK ROAD, HIAWASSEE, GA. (404)896-4342.

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ PAID ADVERTISEMENTS ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

B128 SOFTWARE FROM COMP/TRAC CORPORATION

COMP/TRAC CORPORATION CONTINUES TO SUPPORT THE B128 WITH MOST OF THE COMMERCIAL SOFTWARE PROGRAMS AVAILABLE. CMS INTEGRATED ACCOUNTING SYSTEM $159.95, CALC AND WORD RESULT $94.95, SUPERBASE $14.95, PROGRAMMER SPECIAL SUCH AS THE JCL 700 WORKSHOP $139.95 AND THE 700 ASSEMBLER $99.95. THE APPLIED CALC RESULT MANUAL IS ALSO AVAILABLE FOR $14.95. SEND YOUR ORDER ADDING $3.00 FRT PER PROGRAM TO: COMP/TRAC CORPORATION, 158 HIGH TOP CIRCLE, HAMDEN CT 06514, BROCHURE AVAILABLE, Note: Comp/Trac will be closed for annual vacation from April 24 thru May 14.

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ PAID ADVERTISEMENTS ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Gary L. Anderson
Anderson
Communications
Engineering
1528 34th St. SE.
Cedar Rapids, Ia.
52403

24K RAM CARTRIDGE with Case for the B-128

Assembled+Tested (Socketed Memory ICs)	$39.95
Kit (Case, Board, ICs, Sockets, Caps)	$34.95
Bare Circuit Board and Case	$14.95
Shipping/Handling per order	$ 2.00
Iowa Residents add 4% Sales Tax	

COMING SOON: 1 MEG MEMORY EXPANSION BOARD for the B-128, Banks 0-14, Socketed 256kX1 DRAM, Pin Fields for Future I/O, Socketed Static RAM for $0800-$1FFF in Bank 15, You Can Install in Just Minutes!

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ PAID ADVERTISEMENTS ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

C-64™ • C-128™ • VIC-20™ • SX-64™ • Plus 4™ • C-16™ • B-128™ • PET/CBM™

One disk. 25 business programs, $29.95

Now for the good part:

The **Intelligent Software** integrated home/business/educational package comes on disk or tape at the ridiculous price of **$29.95** [plus five cents for postage + handling]. It is not public domain or home-brew software; totaling over 50 pages of source code, it was designed and tested as a package to take care of all your data processing needs.

Customers write: "*. . . accolades for the authors. This is as slick a deal as I have seen and more than adequate for all except fancy presentations. The best thing is the ease of use . .* "

"*I have come to consider these programs among the most valuable pieces of software I own.*"

The package includes:

Database: A complete multi-keyed fixed-record-length data base manager. Sort or select [using all relational operators:, =, >, <, AND, OR, NOT, wild card] on any field, perform computations on numeric fields. Any operation can be performed on all, or only selected records. All fields completely user-definable. Can be used for any number of tasks, including accounting, mailing lists, inventory control, record, tape, or book cataloging, expense account maintenance, or as an electronic rolodex. One customer writes: "*I am especially impressed with Database, and have used it to replace a half-dozen other 'database' type programs I had been using.*" Even if you use your Commodore for nothing else, this program alone might justify its expense.

Word Processor: A full-featured menu-driven word processor including: very fast file commands, screen editing, text locating and full control over margins, spacing, paging, indentation, and justification. "*. . . well done and highly functional . . . Provides an excellent alternative to the high priced word processors . . . this is an excellent buy. Highly recommended.*" — Midnite Software Gazette. "*Provides good basic features.*" — Compute's Gazette.

Copycalc: An electronic spreadsheet. Turns your Commodore into a visible balance sheet; includes screen editor. "*Excellent program for budgeting, estimating, or any math-oriented use . . . well worth the money. Highly recommended.*" — Midnite Software Gazette.

Also included: **ReportGen, ReportMerge** [these interface W/P with Database to create form letters, statements, invoices, mailing labels, other reports.]; **Baseball Statistician** [compiles batting statistics for a baseball league]; several W/P utilities, including **Index** [indexes W/P's text files]; several Database utilities, including **DBmerge** [facilitates multi-file database applications.], and **DBStat** [analyzes D/B files]; a programming utility, **ASCII,** which converts text files [program listings] into program files; also **Checkbook**; **Inventory**; **Paper Route**; **Loan Analysis**; **Breakeven Analysis**; **Depreciation**; **Labeler**; **File Copier**; more.

Versions of the package are available for every Commodore computer having a minimum of 10k RAM [including the C-128 in 128 mode]. All programs will support tape, disk, and printer. Price includes documentation; Calif. residents add 6%. Add $3 for credit card, COD, 8050 disk, or cassette orders [cassette not available for Plus4™ and 16™.] No personal checks from outside USA. This ad is the catalog; send $1 [deductible from order] for a sampling of program output.

Intelligent Software
Quality Software since 1982

Box A Dept. M-7
San Anselmo, CA 94960
(415) 457-6153

RIBBONS - MEMORY KITS - FANS - POWER SUPPLIES

			12+	1-11
CBM 4023	Multistrike	ea.	$5.50	$6.00
CBM 8023	Nylon Fabric	ea.	$5.00	$5.50
CBM 1525	Nylon Fabric	ea.	$5.75	$6.25
CBM 1526	Multistrike (Carbon)	ea.	$5.50	$6.00
CBM MP801	Nylon Fabric	ea.	$6.50	$7.15
CBM PET	Nylon Spool	ea.	$2.90	$3.20
CBM PET	Multistrike	ea.	$4.05	$4.40
CBM 6400	Nylon Fabric	ea.	$4.95	$5.45
CBM 6400	Multistrike	ea.	$4.95	$5.45

Minimum Postage: $2.00 plus $0.25 each additional ribbon
Please give Street Address for UPS shipments -- Foreign Add 15%

Memory Expansion Kits (each)	($3.00 UPS)	$75.00
Memory Upgrade (In Shop)		$100.00
Fan Kits	($5.00 UPS)	$25.00
B128-80 Power Supplies	($5.00 UPS)	$45.00

S & W Supply Company
Jim L. White
5308 Timberline Trail
Rapid City, South Dakota 57702
(605) 348-3696

***

CompuServe #70007,1147
[Prices Subject to change without notice.]

CBUG SCRATCH PAD

By: Norman Deltzke
February 28, 1986

------ MAN DID WE GROW!

As of February 1985 CBUG had about 2000 members in the US and CANADA plus a few in Australia, England, Germany, Italy, Mexico, Central and South America. I have to say "about" because our database is still not quite up to date, AND new applications are still pouring in. Many of you received your January 1986 SCRATCH PAD telecommunications issue a few weeks late because of the database delays. For this we apologize.

This issue of the CBUG ESCAPE is a double issue because of the heavy flow of registrations, articles and library contributions over the last three months delayed the Winter issue. I had planned the ESCAPE to be a 24 page quarterly publication. Obviously this 80 page issue far exceeded that budget in contents, time, printing and postage. The flow of incoming disks has been most encouraging. Chip in everyone, send in that program, game, application. Give our librarian a real challenge! I've heard from people in all walks of life who have written programs for their own use -- musical instrument diagnostics to classroom attendance record keeping; pig farming records, to medical insurance form execution for doctors and hospitals and the like. Don't sit on them, send them all in. If you needed it, someone else does too! More important, your style of programming might just show up some unique tricks OR your program might be easily changed to another unrelated application saving hundreds of hours of work. I've found that though I can't program at all, I can go in and "butcher" some things to make them useful to me! CONTRIBUTE your programs, project notes, etc.

Item 12384 on the order form is marked as free will donation. I want to thank all of you who have generously contributed, as it is your donations that have made this triple expanded combined issue possible.

We have included a raft of articles on the RS232 port both as to printer and modem interface. Normally, we could and would not distribute so much specialized information in printed form as it belongs on the soon to appear CBUG Encyclopedia disks.... but I put it in this time because, a.) there are far too many members facing RS232-printer interface problems, b.) I wanted to show everyone how a few people can really help out all the others, and c.) give a sampling of the type of materials that are going to be on the encyclopedia disks. Some of this type of material will also show up on topical disks, as in the case of the telecommunications disks CBUG #11 and 11a; at other times in pre-releases to get the materials out to those in urgent need prior to the compilation of an encyclopedia disk.

Are you registered with CBUG? CBUG has not in any way affiliated with any other organization so you must register specifically with CBUG if you want to continue receiving the CBUG ESCAPE. You can easily tell from which mailing list your name came: CBUG members are shown on mailing labels with a CAPITAL letter to the right of the Zip Code (example: "60007 A"). Different codes are from other lists, and are NOT registered members of CBUG. If you are not registered with CBUG, you should do so NOW! If you are already registered, do not re-register. There is a $10.00 per calendar year subscription fee; this is the last ESCAPE sent to those who have not registered with CBUG OR have not paid the CBUG subscription fee. If you've sold or disposed of your B128, please tell us, and to whom it was sold so they may be offered the benefits of CBUG.

With a bit of luck, the entire membership will receive THE ESCAPE at about the same time -- as the database is nearly up to date. As you will note, this is a huge publication, weighing 6 ounces. Please advise us promptly of any duplicates or errors.

------ WHAT CBUG HAS DONE

Locally CBUG has two meetings every month, as explained in full by Marilyn Gardiner in excerpts from her CBUG LOCAL newsletter later in this book. On Sunday, Feb. 23 we were honored by the attendance of Mr. Jesse Knight, publisher of Knight's copy utility. Mr. Knight fielded questions from numerous CBUG members attending.

Many people have sent in the coupons for local zip code print outs of CBUG members so that local groups can be formed. This is dependent on a complete database -- I hope to tackle that project later in March 1986.

Tell us what your needs are and how CBUG members can help you. i.e. Suggest subjects for articles and programs, tutorials and aid files; use the special form provided to make our sorting easier. CBUG will publish the easier and shorter stuff in written form and the rest on disk.

Many members have listed themselves in the CBUG Yell For Help telephone directory. Please send in the Yell for Help registration coupon if you have skills in any area and can spare the time to answer phone questions from members. I have received many hundred calls in the last six months, I can assure you that it is a most rewarding endeavor -- one of the most pleasant parts of running this organization. Remember that we mice in Chicago will be unable to handle all the needs of our growing organization without a willing base nationwide. Please take on portions of this load. With the recent publication of the Yell for Help directory, many members have been able to reach qualified helpers faster and with fewer referals than before.

The CBUG library is our pride and joy. We have some super new offerings. Compliments and plaudits are due to the CBUG librarian, Mark Schwarzbauer -- the several new disks he and his group have compiled for you are exhibits of how a library should be run!

There are a number of copyrighted programs offered thru our library, some on library disks, others are semi-commercial projects such as BeeLine. Please do not make copies of copyrighted materials for other persons -- the Federal Copyright laws allow you only to make copies for your own use persuant to the "fair use doctrine". Many of our authors have devoted a great deal of effort and money to bring you programs that are offered as freeware or even without a request for gratuity. A notable example is Mr. Freeman's programs to manage a retail news distributorshio. These are the result of over one man year of paid programmers time! Please write people whose programs you enjoy, if possible send them a buck or five from time to time. Pictures of deceased presidents always make nice thank you cards. With the exception of copyrighted materials, CBUG disks are offered as public domain. However, we to have gone to considerable effort and expense to compile the information. This is why there is item 12401 Recopy Fee on the order forms. Remember, if you do provide copies to your friends or local club members, CBUG has been deprived the very revenue which allows it to operate. So please send in $5.00 for each non-copyrighted CBUG disk which you copy for others.

Many current CBUG members were also members of an earlier organization based in Lompoc California. In this issue we offer a library disk containing the entire print files of all publications of the deceased Lompoc BUG along with the various programs in its disk library (not elsewhere released). It appears that the Lompoc BUG is no more -- their last publication was dated May 1985; and any of their library orders were reportedly completed by the one remaining "officer" during Sept. 1985. We have been advised that that the operation is closed. As a courtesy, we have been mailing our publications to all of the Lompoc BUG members, though many have assumed their dues to the Lompoc BUG were grandfathered here. This is not the case. With the expense of postage and printing our generosity must come to an end with this issue.

As you read in our last publication, the January 1986 SCRATCH PAD, our presence on the Delphi Network is providing a fantastic help to B128 users. Our Sysop, a local Chicago B128'er, Louis Lohman, is devoting as much as two hours a night giving aid to those who ask questions. Best yet, Elizabeth Deal is now available via Delphi. Just leave your message to Lou (user name = LouisL), mark the body of the message for forwarding to Liz, and Liz may try her hand at your question. All of CBUG owes a great deal to Liz for her programming efforts for the B128, her topical articles, and now her entrance on Delphi. I've no idea how she squeezes us into her busy schedule, but lets all have a round of applause (and thanks) for Liz and her continuing invaluable support. Delphi is a way to get the fastest answers from a whole slew of experts. Others can profit from your questions and the resulting answers. Eventually some of this information will find its way into a Q&A column in the ESCAPE or into the CBUG Encyclopedia disks; meanwhile it will always be archived in the database portion of Delphi. Here is a recent example of what the group on Delphi worked out: The question posed was how to underline on a 4023/8023 in Superscript. This is, of course easily done by using CONTROL 3 on the next line, but often you are forced to use double space and pre-format text using ESC O C S. Our crew on Delphi worked out how to do it without double spacing, using fractional line feeds! We had all thought this was impossible in Superscript!

Delphi will be increasing their evening rates from $6.00/hr to $6.60 on March 3, 1986, HOWEVER, they will grandfather the lower rate for subscribers signing up for the Delphi Advantage program which requires 4 hours minimum per month but also provides you with additional services and publications. The Advantage program may be subscribed at any time thru June 1, 1986

------ COMMODORE/DOCUMENTATION

I have decided to give myself a new title for this paragraph: Chief Turnip Bleeder. In the past, CBM has been most reluctant to divulge anything of value regarding the B128. Thanks to the kind efforts of some members of the CBM staff, and the obvious sincerity of the many CBUG members who have written and called CBM, there is now an operating pipeline thru which critical information is flowing to CBUG. Our Library NOW has a micro fiche containing 302 pages of CBM internal data on the B128 and its related family of models. Memory maps, processor architecture, bus organization information, i/o protocols, etc., even the data sheets for the proprietary processor and i/o devices. Also in our Library are the 8088/8087 co-processor schematics and CCPM86 interprocessor and ROM codes (in printed form). We've just received another box twice as large as the last full of information which will be offered in the next CBUG publication. It also will be released on micro fiche.

Why microfiche? Short run printing is extremely expensive to produce. With micro fiche we can deliver the same information to you at an affordable price. You save so much just on the first three fiche that with the savings you can buy you a brand new fiche reader (they are about $120 new, about $30 to $40 used). Of course nearly all public libraries have readers. If you find a range of pages for which you want hard copy and do not wish to do it at the library from the fiche, we can provide paper copies of specific page ranges on special order.

Enough for Chief Turnip Bleeder....

------ GIGITS, GADGETS, DOO-HICKEYS THAT WERE NOT

We now have TWO members offering 1 megabyte memory upgrades -- as well as a number of sources offering 256K upgrades. Even though there is not yet programming written to fully exploit 1 Megabyte, Superscript users will welcome having 9 banks of 714 lines to work with concurrently (3 banks with 256K). Take it from one who couldn't see the use of 256K until on a lark I had Jim White upgrade one for me. Now I can't do without those extra banks. View/work on concurrent components of an article or letter just by pressing ESC j #; look up some other file, etc. Calc Result will use up to 512 K if available, so for heavy users this is really a prize. Word Result, the alternative professional word processor from Handic will operate concurrently with Calc Result -- both programs on a split screen together, with the ability to simply pop information from one half of the screen to the other. Great for statistics, financial proposals, forms, etc. BUT you must have atleast 256K to run both programs together.

The co-processor projects seem to be moving along at a modest pace. Three members are working on the 8088 board project with one beginning to build a prototype. A fourth member has received the original CBM prototype of the board (albeit in sad condition) and hopes to get it working in the next few months. The 8088 processor was designed to support CCPM86 and generic MSDOS. It sports a 4meg clock and provision for an 8087 math co-processor. It is believed that it will only be 60% to 65% IBM compatible -- but then that is where many of the clones are anyway. It appears that the 8088 will use any additional memory added to the existing B128. We must all applaud the good planning of Messrs. Anderson and King for the 1Meg upgrades. They have worked closely to insure system compatibility as well as carefully studying the co-processor information to insure correct implementation.

----- WHAT WE PUBLISH, AND ABOUT READING IT

Many of che articles in this publication are far more technical than would normal be printed in THE CBUG ESCAPE, however, some like Gary Anderson's are must reading even for novices so as to get a wider perview of what is happening. Even if you don't understand a word of them, try to read them a couple of times anyway. You'll learn some valuable semantics, and few foggy ideas will start taking root in the gray matter. In time those roots will be put to good use, -- and the next stratospheric article you look at will make much more sense! I almost daily tell members to take the books read them forward one after the other, not to worry that you don't understand a word of it. Then read them chapter by chapter backwards, and any other way you can think of; take them to bed, read them upside down, whatever you may, but read and re-read till it sinks in. Start with Superscript and the B128 handbooks, then work forward. Much necessary info is in the SS book and is not covered elsewhere. It's sort of like trying to learn Russian from a novel without either a dictionary or a teacher -- but a whole lot easier. AND the rewards from "beating the machine" are frequent and great.

CBUG wants what you have to say. In due time, almost everything sent in will be published either on paper or on disk. The general interest material will often be printed (subject to length or abridgement); most all else in disk form.

Generally speaking, CBUG will not edit anyone's work. If you send it in, it will be initially published that way -- with your name on it! As disks are organized by topics and reviewed, then repair/consolidation work may be done -- by Mark Schwarzbauer and his crew. Text files will only have notes added; such notes will be contained between double angle brackets -- <<Notes>>. CBUG's general publications will be without lengthy program listings -- who would want to laboriously type in what can be loaded in seconds from disk anyway???

------ PUBLICATION SCHEDULE

THE CBUG ESCAPE, comprised primarily of member contributions, will be published 4 times per year The CBUG SCRATCH PAD, Announcements and CBUG Catalog updates will be published in short form as often as required between the issues of THE CBUG ESCAPE.

------ FUTURE OF THE B128
THE ORPHAN **HAS BEEN** ADOPTED!

As you've all seen from the ads... Handic is now supporting the B128 with Calc Result, Word Result, and the Pet Switch... (a system allowing multiple computers to share disk drives, printers and other periferals. Incidentally, Handic USA has moved and changed their name in the last few weeks. See the information in ANNOUNCEMENTS BELOW.

Precision Software of England has done a complete about face. I learned from recent correspondence and a conversation with John Tranmer, President of Precision, that they are right now introducing upgraded Superscript III and Superbase II at a very attractive upgrade/exchange price and a realistic first time purchaser price. (see the advertising section) Mr. Tranmer has provided us with several valuable articles on Superbase applications which appear in the CBUG Library. I think we can expect some real help in the future from the brothers Tranmer and Precision.

We have a couple of new advertisers in this issue. Member Bruce Faierson, of N.W. Music and Computer acquired all remaining copies of Superbase, Superscript and the CABS suite from Protecto to assure their continued availability to members. Bruce is quite experienced with all of these programs so there will be some real support for his customers. He has a moderate supply of B128 computers, 8050 drives and printers at substantial savings for those needing back up equipment.

CBUG members represent an extremely dedicated group, financially able to up-grade and enhance their systems. This is an attractive market for third party providers. In many instances, an entire program need not be written from scratch, but rather translated from some other system. This becomes feasible for the dedicated B128/256 market place.

Our own "Professor" Warren Swan has written an outstanding course in Basic 4.0+ for the B128. Yup, we even have our own B128 text book in the Library! The local members who have attended this course have raved as to how easy it is to understand; even for the complete novice. It presents important information even for the most experienced programmers. Warren has really out done himself with these course materials -- all local chapters should undertake to teach the course! It is available from the CBUG library, of course. Order Swan's Basic Course, CBUG order #12773. There is more to come from Warren if you all like this one -- Machine Language for the B128, for example.

Before I sign off, I'd wish to thank each of you who have written letters to me and my family, Liz, Warren, Angel, Tony, Fred and all the others. Its sure nice to know our efforts are so much appreciated. But really, the effort is much larger, and the thanks go also to the many devoted contributors who make this publication and the library possible. CBUG was not formed from thin air, but by the hard work of many dozens of dedicated B128 owners.

Norman Deltzke
Publisher, THE CBUG ESCAPE
The Chicago B128 Users Group - International (CBUG).
312 456 8720 7-10pm CST, Wed. thru Sat. Please use our Yell for Help service during the hours indicated in the Yell for Help directory.
Credits: Bindery and mailing prep by Jennifer & Billy Deltzke (ages 12 & 14), and my wife, Barbara who has dug in helping daily with CBUG projects.

* *

-----------------EUROPEAN RESEARCH

There is a great deal of different programming and variations on the B128 in Europe. I ask our European members to research as much of this information and report your findings to our latest project director: Jeff Spielmaker, CBUG European Information Co-ordinator, 2145 Galewood, Wyoming, Mi, 49509. Phone 616 243 5686 during reasonable hours.

Any U.S. members planning a trip to Europe with a few hours to spare, should contact Jeff for possible areas of inquiry matching your itinerary; and those several members who have begun research should report their findings and plans to Jeff so efforts can efficiently co-ordinated.

-------------------DECOMPILERS NEEDED

Several of the important programs used by B128 members were compiled using PET-SPEED. We need to find a decompiler for both PET-SPEED and other compilers and assemblers in popular use so that repairs can be made to the programs. Anyone knowing of a source or better yet having one, please contact CBUG immediately.

-------------------ASSEMBLERS NOW!

In addition to the JCL assembler package, there is a new one, written just for the B128 assembler being announced by member Howard Harrison. There are atleast 3 others either being writen for the B128 or converted to B128 use.

Member James H. Roach, 111 Sherwood Dr., Benton, La 71006 wrote asking for a PASCAL compiler for the B128. If anyone can help Jim out, please write BOTH he and CBUG.

-------------------PROJECTS SUGGESTED/NEEDED

DRIVES: There will soon be need of interfacing disk drives of other non-CBM types to the B128, such as IBM and CPM floppy drives. One possibility lies in operating the 1571 from a yet to be designed interface, possibly modifying the cassette port and writing appropriate software patches. The Cassette port is electronically and mechanically intact (though not tested at the time of manufacture, most are believed to operate). But the ROM code to implement them was removed from the current model in favor of the built-in ML Monitor. The 1571 would give us an easy way to load numerous other formats into the B128 and the coprocessors as they come on line.

USER PORT(s): We heard reports of members connecting to the internal user port and directly driving Centronics printers. Please provide CBUG with articles and detailed instruction files for doing this. Perhaps some speculation as to what can be done with the user port resource would be in order. ALSO, we are advised that there are two "tri-port" I/O (input/output) chips in our machines, but that only 5 of the 6 ports are used. Possibly this sixth port offers entry means for non-CBM drives for example (or a method of interfacing to the 1571).

TYPISTS: We presently need someone (or several to share the project) with excellent typing skills to transcribe some 40 pages of ROM code for the CCPM/MSDOS co-processor. This must be reduced to disk before the co-processors now being implemented can be made to work. The material is essentially columnar, consisting of 3 columns, 5 digits, 4 digits, and 11 alpha-numeric.

EMULATORS & TRANSLATORS: Everyone knows of the 8432 program which causes the B128 to think it is TWO 8032 computers -- and if you have extra banks and the latest upgrade to the 8432 program you can put an 8032 in each of the 15 RAM banks each running a different program. You can then toggle back and forth one program to another. What we now need is a rewrite of the 8432 so for emulation of 4032 and PET machines. It is, however, generally thought that to emulate the 64 is impractical due to the radical differences in the machines. The next real challenge is to develop programs which "copy" and translate all in one. No easy project, but if it were available, we could open up the entire world of programming from other CBM machines, possibly basic language IBM/PCjr, Apple, etc.

IN ALL CASES ABOVE EXPRESS YOUR INTENT TO THE CBUG REGISTRAR, Marilyn Gardiner before starting. Per chance anyone has already done these or other tasks, please advise Marilyn immediately!!!

INFO-DESIGNS/CABS PROBLEMS As an aid to the many members using this accounting suite, anyone experiencing problems or locating bugs in the program are invited to write up the information in careful detail and send it to CBUG. We hope to put together a comprehensive listing of bugs, fixes/circumvention techniques over the next several months. These programs DO work, but take care and try things before trying to go at it full speed. Remember though, that there are many upgraded versions of these programs. Surprisingly, in some circumstances the older versions may be superior.

MEMBER ANNOUNCEMENTS

-------------------LOCAL CHAPTERS

ONTARIO, CANADA. Mr. E. Todd Taylor, Box 434, Russell, Ontario Canada, KOA 3B0. Phone 613 445 5413. Mr. Taylor is desirous of forming a local CBUG chapter in his area, if not spearhead a trans Canadian effort. He writes that he has both technical and programming background. With the substantial number of Canadian registrants, I hope you folks north of the border can take Mr. Taylor up on his offer.

VICTORIA, B.C. CANADA. Mr. Russ Beinder, 1159 Ranger Pl. Victoria V8X 3P5. Mr. Beinder has considerable credentials in both hardware and software both B128 and earlier CBM business frame machines.

COMAL. Comal is yet another language that can be run on many computers. It hails from Denmark, but there is strong support for it from Len Lindsay (aka CaptainComal on Delphi) 6041 Monona Drive, Madison, Wi. 53716. Comal is not currently implemented for the B128 but PET COMAL will run with the 8432 emulator. Warren Swan is persuing the possibility of converting Comal for direct use on the B128, and Jessie Knight of Knights Copy Utility has published a text book on the subject. Should Comal become a reality, it opens up one more valuable avenue to enhance our orphans.

-------------------REPAIR CENTERS

<<CBUG makes no endorsement of repair facilities, rather lists them as a courtesy to the membership. Members should report their satisfaction level by "confidential" letter. We need your help to keep our listings current and accurate.>>

CBUG ANNOUNCEMENTS

New Registration: Omni Tek, Rich Snyder, 1300 Main St., Tewksbury, Ma. 01876; 617 851 4580 $29/hr entire Commodore line.

-------------------SIG (Special Interest Group) FORMATIONS

See article this issue by Angel Matos re special application programming sig.

-------------------REAL ESTATE INDEX APPLICATION

There are a number of Real Estate professionals in CBUG. We have a S&L appraiser, two county tax assessors, and quite a few real estate brokers. One, Les Combs of P.O. Box 391 Greenfield In 46140 is using his B128 to index over 1100 V.A. Repos for sale in the open market. Contact Les both as to his property listings and his useful application of the B128.

------------------REFERENCE GUIDE INDEX

Member Jack Mangold, 2130 Blackmon Rd., Yulee, Fl 32097 offers the following:
A Reference Guide without an Index! That's what the "Commodore <<Protecto>> B-128 Programmers Reference Guide" is.
I have organized and typed an Index and want to offer it to any member. Those interested need only send me a self-addressed, stamped envelope. I would also ask that they enclose a dollar bill for handling.

CBUG ANNONCEMENTS

-------------------HANDIC HAS MOVED!

If you've had trouble getting thru to Handic lately, blame it on the phone company. After several weeks, they've finally atleast referred Handic's 800 number! Meanwhile to add to the confusion, Handic USA has become an independent US distributor, now known as:

ScanAm Enterprises
190 Moore St. Suite 202
Hackensack, N.J. 07601
201 488 7078

Being that it is now their own show, the management at ScanAm will no doubt be even more responsive to the needs of the B128 community -- which will be real hard to do since from all reports they've been doing a superlative job since the present management was installed by Handic.

-------------------STILL URGENTLY WANTED -- RS-232 PRINTER CONNECT INFORMATION:

In our rush to get the ball rolling, I've not been keeping notes of phone conversations. Some members have succeeded in operating printers AND TEST EQUIPMENT directly on the RS-232 port. Would those members please submit files detailing the necessary procedures for doing this. Many many B128 owners are stumbling seriously on this problem. Please prepare your file in Superscript form and send in on disk. We have printed the totality of our printer interface information in this publication.

In that same area, discussions as to IEEE-488 implementations would be of interest. There are a good many members who are using the B128 in advanced scientific applications with the IEEE tied to all sorts of instrumentation. We'd like to hear from you, both with general interest articles, and specific findings for the CBUG Encyclopedia.

--------------------ADVERTISING

THE CBUG ESCAPE will carry paid display advertising from vendors offering products related to the B128. CBUG makes no warranty or endorsement of those products. Any commentary is soley the opinion of the author. Display rates available on request. Want ads are $10.00 per 110 character line for each issue-insertion order. Payment to be remitted with copy/order. Submit to Norman Deltzke in legible form, preferably typed/printed.

-------------------CORRESPONDENCE WITH CBUG

PUT YOUR NAME, ADDRESS, and evening PHONE NUMBER on EACH item!!!!!!

We are at the stage where everything must be put in boxes and batched. Please do NOT dump numerous unrelated subjects on a single page. Rather, write subject specific memos, each on a separate page, each MUST have your name, address and phone numbers. What and how you say it will determine the type and quality of the Q&A response. Please always head the memos with a SUBJECT/CATAGORY word or two. We are infact asking for lots of input, but please do not make us spend time spinning thru confusion or handwritten muck. We are still receiving letters and memos without any return addresses. Once separated from envelope, they can not be individually answered. Please be complete and legible.

Many more difficult questions or those believed of general interest are accumulated and batched to other members, so allow for considerable delays in response. Of course, inquiries as to your membership or order fulfilment errors are investigated and answered asap.

-------------------GRAPHICS & CARTOONS

Any artists amongst us are requested to provide cartoons to add some levity and/or saterical humor to The CBUG ESCAPE.

-------------------PROJECT REGISTRATION

CBUG ANNOUNCEMENTS

With all the effort many members have expended to accomplish important tasks, our library still stands testimony to the duplication of efforts. Since we do not edit things at CBUG prior to final release, many people have done the same thing over and over. This is not always the best use of resources, though it may be a great deal of fun and highly educational. If you are planning a major undertaking, please register your intentions with Marilyn below. This way teams can be established, and efforts best directed. This is a courtesy service and is no way binding or prohibitive of anyone doing it anyway. Multiple solutions to problems are indeed most enlightening. It is thanks to Marilyn's efforts that Messrs Anderson & King have coordinated their memory expansion plans to insure future compatibility.

CBUG PROJECT REGISTRAR: Marilyn Gardner, 1630 Madison St., Evanston, Il. 60602 -- 312 866 9159 7:00 PM to 9:00 PM CST.

------------------YELL FOR HELP TELEPHONE ETC. PROTOCOLS

COLLECT CALLS WILL NEVER BE ACCEPTED BY ANY CBUG HELPER.

DO NOT CALL HELPERS WITH QUESTIONS OUTSIDE THEIR FIELDS OF EXPERTISE.

READ ALL AVAILABLE INSTRUCTIONS BEFORE YELLING FOR HELP.

DO NOT PESTER OUR HELPERS, OR THEY WILL QUIT -- AND WE WILL ALL BE SUNK.

IF YOU REQUEST A WRITTEN ANSWER, PROVIDE SUFFICIENT POSTAGE STAMPS WITH YOUR INQUIRY, AND ALLOW ADEQUATE TIME FOR A RESPONSE. PLEASE DO NOT SEND SELF ADDRESSED ENVELOPES ANY MORE AS THEY RAISE THE COST OF FORWARDING AND/OR MAY BE TOO SMALL FOR WHAT IS GOING TO BE RETURNED TO YOU. THERE ARE NO PROMISES THAT ANY MAIL ITEMS WILL BE ANSWERED HOWEVER AS THEY CONSUME MUCH MORE TIME THAN MANY OF US HAVE AVAILABLE. WE ARE ALL VOLUNTEERS HERE, WE TRY BUT CAN'T ALWAYS DO IT ALL.

WORD RESULT ... A WORDPROCESSOR FROM SWEDEN OFFERS B-MACHINE OWNERS AN ALTERNATIVE TO SUPERSCRIPT II ... WITH SOME ADDITIONAL FEATURES

by Frederick A. Lovejoy

"Yes Virginia ... there really is a Santa Claus!" In 1985, Santa to B-machines users turned out to be Handic Software ... and the gift arrived when they began bringing over from Sweden again after a one-year hiatus ... their excellent wordprocessor, WORD RESULT ... which now sells for $99.95, a substantial reduction from the original price of $195.00.

**** A LITTLE BACKGROUND ... In May, 1984, before Commodore let it be known that they were going to dump all their B-128's in the hands of the good folks at PROTECTO ... who then proceeded to sell the B System for a price that made our machines the buy of 1984 for those who knew a good deal when they saw it ... I learned that Handic, then located in Mt. Laurel, New Jersey had a few copies of Word Result. Sad to say they were all gone before I could order mine. It was not until the summer of 1985 that Handic Software ... now located in Carlstadt, New Jersey received another shipment. My long sought after copy arrived in late August, 1985.

**** WAS WORD RESULT WORTH THE WAIT? ... Word Result is really entirely different than our somewhat flawed Superscript II (Protecto/Precision Software version). Here are a few features for starters:

**** It is menu driven, with a main menu of twenty verbs ... each activated by hitting the appropriate function key, which in turn brings up a sub-menu. All commands, and formatting is done from the menus. The embedded printer commands in the text ... are created via the menu system.

**** It uses a dual drive with the program disk in drive 0 and the data disk in drive 1. This means that Commodore's SFD1000, one megabyte, single drive will not work with Word Result.

**** Word Result includes a simple database ... called a Register ... which allows a maximum of thirteen fields, with no more than 58 characters per field. This can be used for mail merge and will work with Calc Result. It creates sorts using the key field as designated. The key field can be changed allowing additional sorts which can be saved to disk.

**** Word Result can be loaded together with Handic's powerful spreadsheet, Calc Result. You can split the screen with Word Result on the lower half and Calc Result on the top. Data from the wordprocessor can be placed in the spreadsheet for calculation ... and then Calc Result data and charts can be incorporated into a Word Result document. This feature is only possible when you use a B-256 or a B-128 with memory expanded to 256 K. When Calc and Word Result are loaded ... and the split screen is being used ... only seven lines are available on the lower half of the screen for wordprocessing.

**** Word Result provides an optional security feature ... which prevents anyone from seeing a document once it has been created without knowing the password. This is a handy feature when working with sensitive data such as salary information, etc.

**** There are also IBM PC/compatable versions of Word and Calc Result with a nearly identical menu and sub-menu structure and screen format to the B Series. Those who work on both Commodore B and IBM microcomputers could avoid the confusion of jumping back and forth between two very different wordprocessors ... by using the Handic software for both. Transfering files between IBM and Commodore machines would also be made easier it is presumed.

**** Users who MUST have a spelling checker ... and there are plenty of us ... will be mighty displeased as Word Result has none at all.

**** Although the advertising indicates that Word Result can work in eight different lanquages ... the version I received had not implemented this feature. For one who works and writes part of the time in French Canada ... this was a disappointment.

**** The manual that comes with Word Result needs further work ... as some features of the software are not mentioned at all or are left with inadequate explanations.

**** Word Result comes with a Master Program Disk which is used to make a working program disk, a cartridge and a manual. The cartridge allows backup of program disks ... but creates a problem for those who have more than one B machine in use at the same time. In short, you will need a cartridge for each machine. Cartridge failure could be a disaster for those who keep important records and correspondence in Word Result form. I know ... as my first cartridge failed. Heavy Word Result users will want to purchase a backup copy of the program.

**** Unlike Superscript II ... Word Result has no limit to the size of a document. This is very helpful for those preparing long manuscripts.

WORD RESULT SETUP AND USE

Since all but a very few B Series owners have never seen Word Result in operation ... and may have an interest in ordering the wordprocessor ... a brief description of the program is in order.

SYSTEM INSTALLATION ... Setup requires inserting the Word Result cartridge in the B machine prior to turning it on ... then placing the Master Program Disk in drive 0 of your 8050 or 8250 and the load program. Next you must make a backup master disk, at which time you will select your printer(s), listing a maximum of one daisy wheel and one dot matrix machine. Note that the lanquage option does not work. If you have a B machine with 256K and want to use Calc Result with the program ... you must also include this program on the disk. Next step is to create a data disk. At this point you can also configure other printers as required, but you will need to create a new Master Program Disk to do so. All this is described in Part I of the Manual (17 pages).

ONCE THE SYSTEM IS SETUP ... LOADING WORD RESULT IS A SNAP ... Just insert the Word Result cartridge in your computer before powering up ... then insert the program disk in drive 0 (the copy made from the Master Program Disk supplied) and the data disk in drive 1, hit Shift/Run ... and the program along with the file you were working on last (if any), loads in about 54 seconds ... which is about the same time it takes to load Superscript II.

USING WORD RESULT ... With Master Program Disk in Drive 0 and Data Disk in Drive 1 ... you are ready to go. Part II of the Manual is a Tutorial (39 pages) of thirteen short lessons that are poorly written and lack sufficient depth to really do the job very well.

At the bottom of the Word Result screen is a main menu of twenty commands. These correspond to the twenty B Series function keys (function and shift/function) which operate the basic commands (verbs) that control Word Result. Hitting a function key will bring to the command line on the screen a sub menu of noun options ... which activate the various functions. Striking the Commodore key shows the function key that goes with each command. Shift/Commodore key gets you a helpscreen that provides short definitions which are also found in Part III of the Manual (13 pages). An information line above the command line displays the function in operation as well as questions and errors messages. Another line shows the cursor position by line, column and page, as well as file name, the edit mode and mode. In addition the screen contains two rulers. All this remains when the screen is split for use with Calc Result ... leaving only seven lines for text space on the Word Result half of the screen.

Some of the command verbs in the main menu are close enough in meaning to cause confusion ... for instance Remove and Erase, Modify and Change, Get and Display, etc. After working with Word Result for awhile ... I have found that it is easier to develop a Display Card which shows for each function key the respective main command and sub menu with the purpose, steps to take to get results (keys used) ... with occasional notes about problem areas. This saves poking through Part IV of the Manual ... which cover the function keys in 31 pages, or wasting time with the helpscreens.

Word Result is a very powerful wordprocessor that can do a great many things. There is a document information page that contains forty parameters. You can use the default parameters if you wish or tailor them to your needs. No doubt you will want to ... as the default paper length is 72 inches ... showing the programs European origin. Just about any feature you would want can be found and customized on the document information page which when modified is automatically stored to disk. A nice feature is the extensive information provided about each file (name, author, date created and modified, time, and group) which is shown in the Library (Disk Directory) and is arranged in alphabetical order by document name. Word Result features include password protection, multiple printouts by page or document, proportional printing, an abbreviation list (which allows "boiler plating"), hyphenation, underlining, etc.

In addition ... Word Result allows you to create macros ... that is you can program it to automatically carry out a series of functions when activated. Word Result can be used with a hard disk such as the Commodore 9060. Data files can be stored in ASCII format so that they can be read by other commercial programs ... and Word Result can also read data that is in ASCII from other programs.

IN SUMMARY ... Word Result is an excellent page based wordprocessor that is vastly different from Superscript II. It is not easy to learn due in part to the command structure and the less than perfect manual. It does many things that Superscript II cannot ... and is especially useful for those who want to integrate and process spreadsheet information in reports.

Word Result and the companion Calc Result is available from Handic Software Inc., 400 Paterson Plank Road, Carlstadt, New

Jersey 07072. Telephone order number: 1-800-524-0484. It is also available from selected Handic dealers.

<<NOTE, Handic has since moved. See the announcement section>>

This is the first in a series of occasional columns on Word Result. Future columns will deal with specific applications, tips on use and other areas of interest. CBUG ESCAPE readers are invited to send along user experiences, operating tips, etc. for use in future Word Result columns to Frederick Lovejoy, Post Office Box 2651, Scottsdale, Arizona 85252-2651. Contributors will be acknowledged in the column if their material is used.

ABOUT THE AUTHOR: Fred Lovejoy is President, Frederick Lovejoy Associates; Managing Director, The Princeton Research Institute; author of six books on corporate development; lecturer; consultant on corporate policy and mergers & acquisitions; a former university professor; and Program Chairman , Acquisitions and Mergers Week Conference since 1966. He has started an owned six businesses. BA ... Wesleyan University; MBA ... Wharton; Doctoral ... New York University. He is a B Series enthusiast and the proud owner of three High-profile CBM-256-80 systems. <<The factory original 256K version of the B128 with built in hi-res monitors.>>

AN INCOME TAX PROGRAM FOR THE B128

by: Col. J.E. O'Halloran

Since I had used the C64 for several years to do my income tax, I began making queries of the producers as to the possibility of them writing a program for the B128. I wanted one badly but could elicit no enthusiasim in the software firms, SO-O-O I decided to develop one myself.

I had already purchased an update to the program I had for the C64 (it is for sale cheap and hasn't been used this year) when I started on the B128 but as I went along I tested my program by using figures from my records (also kept on the B128 through-out the year) to prove each segment and by the time I had the program together MY TAX FILING WAS COMPLETED.

Any one who has become familiar with the use of a spreadsheet will be off and running as soon as the program has been loaded. Those not familiar with spread sheets will find a printed HELP sheet to get them going and numerous HELPS built into the program. The Function keys, the C= and CTRL keys all play important tolls in finding HELP and using the program.

Note: These programs operate within Calc Result ONLY!

The formulas for calculating and moving figures to other forms are built in AND protected from all but the most determined effort to destroy them. In several cases where there might be a choice of figures from two lines to be used---the program makes the correct choice for you.

Those who have a spreadsheet program can easily set up formats to use in keeping records of all expenses, investments, interest income, etc., which when entries are made on a regular periodic basis will at TAX REPORTING TIME make filing preparation a matter of an hour or so instead of several days with papers, receipts, memos, etc., scattered over the kitchen and dining room table. Of all the utilities I have used on my computers there are none which give the gratification that I get from having my Income Tax Filing practically do itself in minutes.

Beyond a TAX PROGRAM there are many other very useful utilities which I have devised and keep on disk for quick use when needed. I have programs for figuring the monthly payment on any amount, for any rate of interest and for any lenght of time. Another for the sum of the digits which is very useful in calculating depreciation or Rule of 78. I have an amortization program also for any variations of time, principle or interest rate. All calculations are made immediately upon entering from ONE to THREE figures in the designated places. All print out for permanent record. I find myself using them very often. I have several desk calculators but usually I prefer to use the B128.

Norm and others have said many times that WE, the B128 USERS, must depend on ourselves if we are to realize the potential of our B128 wonders. They are absolutely right. Even software manufacturers (who made what we have been able to get) are of no help. As one of them told me over the phone a while back, "none of us here use that program (one they put out) and we know nothing about it". We have all had to learn on our own OR with the help of other B128 users. Hence, we MUST NOT let this USER GROUP fall by the wayside as the one in California did. Norm Deltzke, his wife and children deserve our admiration and thanks PLUS our moral and financial support. Others who have so generously contributed in time, effort and programs are also to be commended.

Col. J.E. O'Halloran II   RT.2 OWL CREEK ROAD, HIAWASSEE, GA. 30546

ABOUT THE AUTHOR: Col. O'Halloran was raised in the sand hills of Nebraska, where upon graduation from high school at 16 entered the family hardware, lumber and implement business. With 9 years of flying lessons, at 24 left to persue a flying career. First as an advanced flying instructor (Spartan at Tulsa), and on to England to fly for the British in 1941. After two years Jim transfered to the U.S. Navy seeing duty in the Pacific theaters. Then back to Germany for the 1948/49 airlift.

On to the civilian world, a BA in business administration from Ohio Christian, and extensive training in insurance and estate planning, real estate, investment counseling, etc. Before establishing his successful Estate Protection and Investment counseling business, Jim held several positions such as Military Products and Sales Manager of a multi-billion dollar firm, and Executive V.P. of a high technology firm. For a while he went back to flying as a Captain in an international airline.

An avid horseman since the age of three, Jim is just now winding down an 11 year enterprise breeding prize quarter horses. Jim is a SEC lisenced investment advisor, as well as holding numerous other related lisences.

Five years ago Jim went into a self inflicted saturation course on using word processors, spreadsheet and financial utilities. Jim is now a devoted B128 user -- and expert!

THE PENNSYLVANIA CONNECTION

by: Elizabeth Deal
337 W. First Ave.
Malvern, Pa. 19355

Needs change, ideas change, programs grow in response to new needs and challenges. So, I am submitting updated versions of some previously written utilities, as well as just a couple of new programs.

Copy-all program had an obscure bug that wouldn't copy relative files beyond one record. The fault is mine, not Jim Butterfield's. More details in the article, below.

The most notable changes in Supermon are: I've replaced a test for STOP key by my own code, so that STOP key no longer disconnects the CMD device. I coded somewhat better storage verification - this is useful to people who keep changing memory chips in the B-machine. P command now also functions similarly to D, in that you can screen edit the outputs if the output device is screen. Finally, for those who have replaced CALCRESULT chip by memory, SUPERMON's bank 15 hook can be placed at the very tail end of the $7000 memory block. The loader detects where you have memory.

The important changes in Keytrix are:

1. A brand new RENUMBER command - we now have a 4-parameter-renumber, which allows renumbering any range of lines using any starting line number and increment (so long as numbers don't clash, of course!).

2. We can now tokenize listings into programs. This includes listings made on the CBM machines or other computers.

3. You can set up a temporary Basic partition at the end of the current Basic text for a trial run of a second program, or whatever purpose you can dream of. Users of Brad Templeton's POWER on the PET and C64 know this feature as the TEST and BACK commands.

4. Keytrix bank 15 hook can now go at the tail end of Bank 15 memory if you have replaced CALCRESULT chip with RAM. This little piece of code abuts Supermon's hook, as before.

CABS file reader (reader/writer.3) now includes reading and printing INVENTORY files, in addition to the files it previously handled.

Pre-superscript program (bounce fix) has one byte changed. Unless you play a lot with the priorities of interrupts, you won't know the difference, but to be safe, version 3 is corrected. This program might now run in three versions of the B. I haven't tested it, but am basing the changes on the ROM dumps several people have sent. The changes have no effect on the existing users.

Incidentally, any of you who have version 6 of keytrix should make a correction in the loader by adding one line: 182 BANK 1:POKE 60328,6. Then resave the program. It fixes a bug in the GETL command which did odd things to programs made from listings if the listing contained "ready.". Sorry about it.

APPLIED CALC RESULT

This is a review of a book which can be a great companion to the B128 Calc Result program. The book is actually for the C64, but that makes little difference. The disk has B-128 specific spreadsheets (I haven't seen the disk, though, but the card inside the book includes CBM 700 B/BX series, and that has to be us). The book is written by Erik Fagerland, the title is "Applied Calc Result" and it is published by HANDIC.

In spite of an excellent tutorial, Calc Result isn't simple to learn. The purpose of this book is to provide ready-made spreadsheets for various applications, from home budgeting to figuring an area under a curve. You can type in the models following the listings in the book, or you can buy a disk <<see NOTE at end of this section>>. Disk's price seems a bit stiff, $50, but that comes out to $2 per spreadsheet, clearly worth it, even if you were to use just a couple spreadsheets. I don't know the price of the book, I'd guess it will run around $15. Check with Handic.

To me the major strength of the book is in its beginning section which summarises Calc Result commands much more elequently than in the manual, which I find too long and to hard to flip through. This section is well worth having the book. The 25 spreadsheets are then described - there are screen illustrations as well as some instructions about entering the formulas and the data. The final section of the book, unfortunately, contains rather unimaginative "spreadsheet building listings", for lack of a better term on my part. Unimaginative, because it lists every single cell - its value or its formula, instead of showing the major formulas and means of replicating them (after all, isn't that where the power of Calc Result is?). So, with the exception of couple models which contain only a dozen or so lines, the remainder are pages and pages of cell-listings. Not very educational, though excellent typing practice. Getting a disk seems like a very good idea!

Most of the instructions in the book are good and clear. The book has lots of typos and misspellings, but none that would affect the work. Several (one or two) instructions do not work on the B computer as described in the book. However, since the book says what they're trying to do, when things do not work, you can look up the solution in your main Calc Result Manual. Several sentences leave me rather baffled. For example, "It should then be possible to summarize the income and costs monthly use SUM function in CALC RESULT.", but even those aren't hard to figure out.

Overall, this is a good book, worth having, despite several silly flaws that it has. But if you plan to use the models which are in the book, and if you don't think you can figure out what the general formulas are, you better also get a disk, otherwise typing models 18 or 19, for instance, is doomed to failure, I assure you. I think the book is really a set of instructions for the use of the disk.

<<NOTE: Problems re Handic's Calc Result application disk. This disk is not yet available in the US, and is currently only produced for the C-64 verions of Calc-Result. Handic will be furnishing CBUG with an examination copy of the C-64 version for purposes of ascertaining the portability of the disk materials to the B128. The disk is in 1541/4040 format which is NOT readable on 8050 drives so the first step is to copy the data to 8050. Then there is the question of the 40 column format used in the 64 and the possible problems of conversion. CBUG will report the findings of these experiments as quickly as possible, but it will take several months.>>

KEYBOARD SHIFT-L BUG

There is an elusive, though rather minor, bug in the B-machine keyboard decoding. It only affects couple keys, no more than five, I think. One of them is shift-L (as in letter). Shift-L is useful to load programs with, when you can abbreviate DLOAD as d, shift-L, filename in quotes, drive and so on. The abbreviation always works in text mode (lower case/upper case) but it sometimes fails in graphics mode (upper case/graphics), where instead of getting the graphic for shift-L which should be a lower-left hand outline corner (you see on the left square bracket key), it comes out as a big fat X which is on L (should be on V). And that's the mistake - looks like CBM changed the keyboard from the PET days, but didn't tell the software about it.

A little snooping reveals that the graphic mode decoding tables aren't right at all times, specifically if you have gotten into the graphic mode via printing chr$(142) or pushing NORM key or printing rvs-shift-N. There are two solutions to this obscure nuisance:

1. When shift-L fails, use control-left-square bracket instead to be able to load programs (a tip from Howard Harrison in Philadelphia), or

2. If it's important that all keys function as they should, this magic incantation fixes things: BANK 15:POKE 204,0 and it should really be used in programs which switch the character mode using any of the methods mentioned above. (The system leaves a value of 16 in location 204).

If you get into graphic mode on exit from Supermon, the bug doesn't occur, because Supermon leaves the character setup alone as the chips want it with no further changes - and that's one reason why the bug is so elusive. What matters is only HOW you got into graphics, not that you ARE in graphics. Goofy.

HINTS ABOUT COPY-ALL PROGRAM

If you have version 3 of Copy-all program, you may want to fix the relative file bug. After a file "+copy-a*" loads, insert this code somewhere in the program: BANK 2:POKE 7599,168 (to replace TYA by TAY in machine code parlance). Version 4 of Copy-all program has this bug fixed, plus several cosmetic changes, but otherwise it's still the same thing.

COPY ALL can copy all unprotected files. Unrelated to the bug reported above, REL files have to be handled with care: if it's the BIG type, then for it to copy (using COPY-ALL or the system's normal disk command COPY) the disk must be in the BIG file mode. This really calls for careful naming of the files. I prefix the big files with "BR." letters, and normally EXCLUDE them from copying until the disk is switched to the BIG mode. I wish a smart drive like the 8050 could do the switching based on what it sees! The penalty for being wrong is pretty severe: bad block count, and the copied file reports zero blocks.

The remainder of this section tries to answer several common inquiries by showing assorted features which aren't clear from just the prompts of this program.

If you wish to set up COPY ALL, but don't want to run all of it right away, or if you need to quit, just give wrong information at the beginning. For instance, if you ask for source AND destination unit/drive to be 8,0 the program quits.

Another easy place to quit is during directory scan, here pushing Q aborts. During rearranging the order of files, STOP key works. The only dangerous place to quit is after the copying has begun. If you push the STOP key here, make sure to close the files by typing: CLOSE14:CLOSE15, or just plain DCLOSE.

COPY ALL was originally written by Jim Butterfield as means of supplementing the disk's native COPY command. Disk's COPY can only do a file at a time and it cannot cross device numbers. So there is no way to COPY a file from one disk unit (device) to another, if you're blessed with multiple drives (if you are, use CHANDE DEV# program first). Jim's COPY-ALL always included means of selecting files which you do or not want copied. This is done by answering Y or N as the file names are being displayed on the screen. An alternative way, if you plan to copy a group of consecutive files, is to answer N until you see the files you want, then push Y and lock that answer by pressing the RETURN key.

If you plan to copy everything and/or see all the files names on the source disk, just push RETURN over all the questions. The screen prompts contain "default" answers just for that, the answers are asterisks. Asterisk means "go for all". That's the simplest way to use COPY ALL. But we can do more.

COPY ALL allows the user to specify an overall pattern-matching scheme. The syntax is the same as described in the disk book. To copy all files which begin with a "w.", answer "w.*" to the PATTERN question. To copy only 3 letter files, the PATTERN is "???". To copy files named 1test, 2 test, 3 test, your PATTERN can be "?test" or "?te*" depending what else is on your disk.

After all the "matched" names go through, on the B machine you get another shot at rejecting more files, this is done through the D command in the rearranging portion of the program. The actual copying begins after you press C. If you do not wish to use the rearranging option, just answer N to the "want to rearrange" query.

COPY ALL on the B permits you to go after more patterns. Firstly, in addition to specifying the pattern you want to go after, you can specify the pattern for files you just do not want to see. For instance, you can say: show me all files except files which begin with letters "pix.". To do that, answer "pix.*" to the EXCLUDE PATTERN query. This can save lots of time if your disk happens to contain lots of PIX files. Sorry, only one EXCLUDE pattern per run, and you cannot use the "?" character for matching, since we aren't matching. We're only excluding here.

Secondly, you can ask the program to not even bother you about any files, until it sees a program that you want to start at. It invariably happens that you have 100 files on a disk, but really want to copy only last 10. Going through the whole directory is a nuisance. But if you specify "finalversion" in response to the FIRST ITEM query, the program will skip all the disk files until it hits the "finalversion". Then, COPY ALL will behave as it normally does.

Finally, the rearranging section. It's similar in use to Superscript (bit slow and clumsy, but it does the job). Using the R followed by the cursor keys you can range and transfer (X) groups of names around into a desired order. You can grab the NOCOPY "name" and place it as a diveder between the do and don't copy groups, you can delete names, or bring them back ... use your imagination. When everything is arranged in order that you want, push C to actually begin copying.

I like to put dividers made up of some variant of a dashed line, between logical groups of files on a disk. How is it done? BSAVE or monitor-save NOTHING. COPY ALL program contains a line 9000 at the very end. If you look at it, change it to suit and GOSUB 9000, you can make several dividers on your source disk. Then the arranging section of COPY ALL can help you put them in places where needed.

SUPERMON and the PRINTER

The most popular question about SUPERMON is "How can you print the output from the P command to the printer?". I didn't mean to keep it secret, honest. I assumed that everybody knows this from their PET or C64.

Open an output file in Basic, as you normally do for listings, typing everything in one continuous line, no matter how silly it looks:

CBM printer:

OPEN 1,4:PRINT chr$(147);:CMD 1:now push function6 to break into monitor. (You can type CLR-screen for chr$(147), it's simpler).

CBM disk:

```
OPEN 1,8,3:PRINT"0:outputfile,s,w":CMD1:function6
```

Push RETURN and off we go.

After a bit of noise, type your P command, or D or M or ?. If you use the "?" command (M plus Ascii dump) remember to push keys to keep going!

When the cursor comes back, it's likely to fight you. Fight back by pushing SHIFT-RETURN and don't worry about it (CBM bug). Then type X to exit the monitor, and, very important, especially with disk, you must type this EXACTLY:

```
PRINT#1:CLOSE1        <--- very important ending step *
```

In Supermon v5 and earlier, if during P or M or D you use STOP key, CMD may disconnect. If it happens, go through the X and the closing sequence, and re-open this or next file, and start all over. If you didn't use STOP, you can, in most instances, continue with the next bunch of addresses or a different command, such as you may need in intermixing Ps with Ms.

In Supermon v6 (or higher) I've substituted my own code for the STOP test built into the B machine. Pushing STOP no longer kills CMD.

Why isn't printer output directly accessible from the monitor? Partly a force of habit, Supermon always needed the file opened from Basic. The most important reason in my mind is that printers differ, and the monitor would have to be able to handle all of them. There are different commands for spacing, for mode (text or graphics), paging, and so on. It's simple to control all that from Basic - you can customise the output to what you want, issue CMD and it should work.

Incidentally, disk files are neat, since you can later process them, or format neatly for output via Superscript. The stuff on paper tends to be less versatile. BUT do not EVER, EVER forget to do the (*) ending step. If you do, you'll have an * file. Don't scratch it. COLLECT the disk instead, else you could create a mess.

LEARNING MACHINE CODE on the B

Don't if you can avoid it. B machine is difficult to work, more difficult than any other CBM machine. If you must, try to work in bank 15 until you really learn the language. Running code in bank 1 or 2 is a bit tricky. Supermon's Disassembly command is a good source of learning, and practice using the tiny assembler (A-command) until your code gets too long.

You can learn machine code from Jim Butterfield's book "Machine Language for the Commodore 64 and other CBM computers" published by Brady/Prentice Hall. The information about the B machine is skimpy there, but there is more of it than anywhere else. And the book has got to be THE BEST Machine Language course ever written!

LEARNING BASIC

B machines have evolved from the PET and the C-64 computers. Any book dealing with Basic or Basic 4 is valuable. The best PET book is Raeto West's Programming the Pet/CBM published by COMPUTE!. Then there is Jim Strasma's CBM Guide, published by Osborne - I don't have it, but have seen it, and it's good. Sorry I can't quote the full title.
<< PET PERSONAL COMPUTER GUIDE, Adam Osborne/Jim Strasma/Ellen Strasma, Osborne/McGraw-Hill, 1982
PET PROFESSIONAL COMPUTER GUIDE, Adam Osborne/Jim Strasma/Ellen Strasma, Osborne/McGraw-Hill, 1982 >>

For charts, summaries and lists of commands, memory maps, and so on, nothing beats Karl Hildon's "The Complete Commodore Inner Space Anthology" - order from TRANSACTOR, 500 Steeles Ave, Milton, Ontario, L9T 3P7, Canada. The cost is $14.95.

SUPERSCRIPT,
PROGRAM LISTINGS,
KEYTRIX

Here is an answer to a popular question, "how can you bring in program listings into Superscript?". Quick answer: read Superscript instructions, page 94 about INTERFACE TO OTHER FILES. It goes like this in Basic:

```
OPEN 2,8,3,"0:listingname,s,w":CMD 2:LIST <optional line range>
```

Push return, and when the cursor comes back DO NOT FORGET TO WRAP UP THE FILE by typing:

```
PRINT#2:CLOSE2 (no space between PRINT and #, please)
```

Bear in mind, that all graphic characters get lost in the translation as Superscript doesn't honour them. It doesn't matter much of the time. Often you can reconstruct what things meant (S=clear screen, s=home, q is down etc) from the context, but programs containing lots of real graphic characters and/or lots of print-control graphic characters are hard to deal with.

Explanation of some of the numbers: "2" is Basic file number. Use any number 1-127. OPEN and CMD must use the same number. "8" is device number (disk is normally 8, screen is normally 3, printer is normally 4). "3" is a special number inside a disk unit. It's a buffer number which holds data. It's also called a secondary address or a channel number. Use any number 2-14. Reason? Zero and one are reserved by the disk for saving and loading, 15 is reserved for error messages (READ ERROR) or for sending commands (not data) to the disk (SCRATCH, COLLECT, etc).

Incidentally, if you use KEYTRIX, version 5 or higher now contains a command that can be sometimes useful: GETL. This one can take a program written in the Superscript format (or any ASCII listing, even from a foreign computer) and can "tokenize" it into a program format. It may, or may not run, depending where it came from, but the feature can be useful for various aplications.

Also, if you use KEYTRIX, you should know that you can read, in Basic mode, any sequential file (e.g. Superscript created) using the TXT or TEXT command. Versions 5 and higher use slightly different keywords: no longer TXT or MRG, we now type TEXT and MERGE, it's easier.

SUPERSCRIPT's COLUMN MOVE

My level of respect for Superscript has leaped upwards recently when I learned to use the column move command. This has got to be the most powerful command I've ever seen. To do column move you have to define a range using ESC-SHIFT-R, cursor over to the last character to range and down to last line; push return and move it using the cursor keys; push return when all done. Two tricks: (1) do not try to define a range over 255 lines at one time or things will get mangled, (2) be very careful to write yourself a note as to the best path and destination (line, column) for the move. If you don't, you may well wipe off some text you'd rather see intact.

I know people who use Superscript to edit files received via BTERM. Those files often (always?) contain line-feed characters at the beginning of every line. They show up as a column of j's followed (often) by a space. Column move permits stripping the unwanted line-feed junk. You may want to start Superscript with 81 columns originally to handle this mess.

CLEAN PRINTING

Remember when you got your 4023 printer and you were subjected to such gems of instructions as "lift front of printer unit and make it stands as the bottom of case be vertically face to you" and "gently back the unit to lay flat on a firm surface, position the printer be face to you"? What they didn't tell ME is how to insert the ribbon correctly. For months everything looked like a smeary mess, until I saw a note in the TPUG magazine which shed some light on the subject. Here is a picture (top view) for those of you who are as clumsy as me in handling the ribbon:

```
                    o                        o
                    -------------------------
                        metal over a roller
                    -------------------------
          ---  small metal plate attached to the print head
       ----------- here comes the ribbon, NOT NEXT TO PAPER,
             ---                             OF COURSE!!
                    !   !   this is the print head
                                 ---
                              ---------
                              cartridge
                              ---------
```

NO HIGH RESOLUTION GRAPHICS ... YET

Several people want to know how to do high resolution graphics. Sorry, CBM isn't hiding anything in the books - it can't be done. The highest resolution we can get is 160x50 using quarter-graphic characters, as you can see on keys 7,8,9,0, and -. Then, in horizontal only direction you can get 8x resolution using the skinny lines on keys 34erdfcv and in vertical 56tyghbn.

To do real bit-mapped graphics we would either have to hook up something to the existing CRT controller or design a custom board in one of the banks, using, perhaps, the C64's VIC chip or the Plus4's TED chip.

I hope somebody out there knows how to do it. I would very much like to work HR graphics on my B machine without the usual troubles of colors and other such, as on the C64. Perhaps we could talk Hal Chamberlin of MTU to make a board for us as he did for the PET?

SAVING ROMS

With Protecto emptying their entire stock of the B machines, some customers are getting very old versions. To help us identify why programs aren't behaving the same as on most B machines, it helps if we get the contents of your machine. Some information on the subject appeared in the last newsletter. Let's just use one common method. We'll use the built-in monitor and we'll assume you save on device 8, drive 0. If you wish you can also include serial number in the filename, and abbreviate the names to S,B,K (System pages, Basic interpreter, Kernal code).

```
enter the monitor: BANK 15:POKE 6,0:SYS 6 and return
now type:          S"0:SYS",08,0F0002,0F0400 and return
then type:         S"0:BAS",08,0F8000,0FC000 and return
finally, type:     S"0:KER",08,0FE000,0FFFFF and return
exit the monitor:  X and return
```

Don't worry about the last byte, we'll get it from another place.

It's clear now that questionable areas ($400-7ff) aren't used in any of the B versions, and that the location of interface chips is the same, so there is no point ever saving as much as we thought was needed originally.

It appears that a byte in bank 15 at 65529 contains 1 (one) in most B-machines. Some machines have a zero and some 170. BANK15:?PEEK(65529) will tell you what you have.

IBM EMULATOR?

Several people wondered if the B machine can be made to emulate the IBM computer. Anything is possible, but part of the difficulty is in disk format. It is not enough for a computer to have IBM commands or MS-DOS commands; we must have a disk that is compatible with those machines. Commodore's DOS is inside the disk ROMs, not in the computer, hence rough to change. It can be done if you really insist (a perfect example of customised format is the 8050 PHYSICAL EXAM from CARDINAL SOFTWARE!)

A bit of perspective: It's a fad to go IBM. By virtue of being so big and well known, IBM is imposing its own standards on the microcomputer insdustry. But that doesn't mean that their standard is superior. Apple, Commodore, Radio Shack, Texas Instruments, and Atari computers were doing just fine before IBM entered the scene. Each, in its own, peculiar, way added something great and absolutely, unquestionably, unique to their computers. The buyers had a choice of features. This seems to be no longer true. The spirit of discovery and innovation has diminished since IBM overpowered the market. And most people don't even realize all the nice features they could have because all they see is IBM-this and IBM-that. Fooooey.

My B-machine has a keyboard full of characters and print-control features. Few computers can match almost 255 key-variations. Few computers have a built in machine language monitor which permits snooping and changing programs. Few computers have the system software built into ROMs. None can match the ease of editing what you type. Most computers behave as if they were some medieval teletypwriters - every time you make a mistake you have to retype the same old stuff.

Who'd want that? Few computers can match the incredible speed of the B-machine. Some are so slow, one wonders if the code inside the chips hasn't been written in Basic.

I'm certainly glad I have an opportunity to work on the CBM equipment before it, too, becomes like IBM. What I got out of working all the CBM machines is lots of knew knowledge, respect for people who write tiny, compact, efficient system routines in tiny spaces and an understanding of how these beasts function by being permitted to play with its insides. Mainframe-type computers do not allow any of that and are much more difficult to use.

The big question in my mind is "can IBM emulate the B computer?".

RS-232 in Basic

RS-232 interface works just fine in Basic. Some people have connected serial printers to the B, others have used it for modem communications and still others, to move data between computers. The interface can be worked in Basic. The grey guide, on pages 137+ (appendix N) and the Protecto guide, on pages 120-127 show several examples. I can show an application I've used: modem and moving data to and from Commodore's Amiga computer (see B128<-->Amiga program on disk):

1. CABLE. The cable that came with the modem from Protecto was all wrong. So I built another that can be used for two different purposes. At each end of the cable connect pins 4 and 5 together into a loop. Then connect pins 6,7,and 20 into a bundle. Those pins don't go from one end of a cable to another. The only lines that go through are on pins 1,2 and 3. Pin 1 is simple, it just goes from one end of the cable to another. My pins 2 and 3 are on a switch: for modem, a wire connects pins 2 and 2, and another wire connects pins 3 and 3, a parallel connection. For computer communication, we need to cross lines 2 and 3, so that 2 on one end of the cable goes to 3 on the other end. Likewise, pin 3 on one end gets connected to pin 2 of the other end. That's why a switch is handy: it can connect in parallel or let the two lines cross over.

2. SOFTWARE. Modem software does the job for you, but if you need to write something different, the examples in the books can be a starting point. Basically, you open a device just like any other in Basic, but it must be device #2. You also open a secondary address, and there needs to be 4 characters included in the OPEN statement, the first two are important, the last two aren't, but must be there:

```
OPEN 1,2,1,CHR$(CR)+CHR$(CMR)+"++"
     ! ! !       !        !    !
     ! ! !       !        !    '-- 2 dummy bytes
     ! ! !       !        '-- command register byte ($d002)
     ! ! !       '-- control register byte ($d003)
     ! ! '-- secondary address (1,2,3 or 129,130,131; n/y ASCII)
     ! '-- device# always 2
     '-- file# (any 0-127 or 128-255, no/yes extra linefeed)
```

Opens file #1 and will not force a linefeed character with each carriage return. If the file number is greater than 127 then the LF character is also sent.

Device is 2, an RS-232 interface. Basic automaticaly allocates a buffer, which is not overwritten (unlike in the C-64), so you can OPEN the file anytime you wish. You also don't have to worry about closing the buffer too early, Basic handles all the chores nicely.

Secondary address of 1 means to transmit only. To receive, you'd use #2, and to transmit and receive you'd use 3. You would add 128 to the number if you also want a conversion to-from true ASCII.

CR byte is a control register. The settings depend on what you are doing. The settings are: speed in bits/second (50-19200), word length (5-8 bits) and number of stop bits (1.5-2). On page 122 in the Protecto guide are all the values. Let's take a simple example to send at 9600 bits/second (value 14), 8 bit word (value 0), 1 stop bit (value 0) for a total of 14. If you needed 1 stop bit AND parity, the last swith would be on, for a value of 128 (that's the meaning of bit 7 notation).

CMR byte is a sum of handshake, duplex and parity switches. In a setup such as this, we have a 3-line transmission, so bit 0 is off (value 0). Then we should set bits 1-3 also off (value 0), Then comes the duplex mode - on for half-duplex and off for full-duplex for a total of either zero or 16 (it's bit 4, and 2 to the 4th power is 16). Finally, bits 5-7 control parity - any value on top of page 123 can be used, depending on protocol you need.

Once the channel is opened, a normal PRINT#1,A$; or GET#1,A$ outputs or inputs bytes to/from the RS-232 device. When the job is done, CLOSE1 cleans up the house normally.

Page 126 summarises two example settings. If you go through the examples you'll see that the setup can accomodate lots of needs.

PROGRAM FILE TYPES

Many programs on disk can be loaded and run and they produce immediate results. But there is a set of programs, which I loosely call "utility programs" - they seem to do nothing and I get phone calls about a program not working or reporting a syntax error.

Take an example of the SPOOLing program. I consider SPOOL to be a utility. When you load and run it, the only thing it does is sets itself up so that you can use it later, AT ANY TIME. The point is to have this utility available in the machine. When you're ready to SPOOL something, you can open the text file, issue the SYS command and go. The program is documented in a superscript file W.SPOOL. The sys sommand is also documented in the program just above the disassembly, just above where it conveniently reports syntax error. The reason the setup is as I have done, is that I do not know under what conditions I may want to spool - it may be in direct mode, or it may be inside a running program - all the SPOOL does is get ready. How you use it is up to you. If SPOOL is used in a running program, you have to make sure that it is called and used when the IEEE bus isn't used for anything - I can't know all programs to which SPOOL is attached. It is up to you to use it at a reasonable time.

Similarily, Supermon and Keytrix seem to do nothing. But once again, these are utilities. They set themselves up, and you can use them when YOU need them. There is no input, output or any immediate action when you say RUN, other than reconfiguration of the machine and loading of the necessary machine code modules.

PAGING THE 4023

by: Roy Sherman

Just a little follow-up on paging when using other than 6 lpi (lines per inch). (the default mode)

To set the spacing between lines (bottom of one line to the bottom of the next) type: open6,4,6: print#6,chr$(x): close 6 <return>. To determine the number for x in the chr$( ) divide 216 by the lpi desired. ie, 216/9 = 24. Simply substutite 24 for the x above [chr$(24)]. Nine lpi, by the way, wiil allow the printer graphics to connect.

Next we set the number of lines per page with: open 4,4:print#4,chr$(147) <return> - turns on paging - open3,4,3: print#3,chr$(p): close 3 <return> - sets number of PRINTED lines - print#4,chr$(147): close 4 <return> - enable paging again.

To determine the value of p, multiply the lpi by 11 (for 11 inch paper) and subtract 6 for the three blank lines at the top and bottom of the page that the printer is going to put in automatically. ie, 9 lpi X 11 inches = 99 lines per page - 6 = 93 PRINTED lines per page. Therefore p = 93.

To eject a page type: open 4,4: print#4, chr$(19) chr$(147): close 4 <return>. This will perform a "top of form" from within a program the same as the paper advance button on the printer.

Roy Sherman 5013 Main St. Skokie IL 60077 (312) 673-5094

PREVIEW OF CMS ACCOUNTING SYSTEM

by Carter S. Pawlus - 414 Erie Ave. - Sheboygan, WI 53081

Sold by: Comp/Trac Corporation - 158 High Top Circle - Hamden, CT. 06514

Contact: Al Tingley at 1-203-397-3104 for descriptive mailing of this and other programs offered. (Al is a CPA who evidently set up the system for the b128)

Contact: Chuck Stewert at 1-214-289-0677 (Original program author) for any technical program questions.

Preview:

The system program comes on one disk and consists of a General Ledger
Accounts Receivable
Accounts Payable
Payroll

Each program can be run independantly as an entity to itself or can be merged or posted to the General ledger. In all of the programs you are led through each step by highlighted prompts which makes this system really easy to run. Most reports (and there are an abundance of them) may be printed to the screen or printer and started or stopped at any time. A real boon to the novice.

The System menu directs you to the programs plus disk utilities option for making backups of the original program, formatting file disks, copying files, initialization of ledger, copy chart of accounts. There is a general information file for entering your business info - name, date, password (if you need protection), tax numbers, payroll numbers, check numbers, overtime rates, ar/ap aging periods, finance charge percent, finance period, next accounts payable check number, and a group of account numbers to be entered for automatic processing after the chart of accounts is set up. There is also a printer code for highlighting your final printouts if desired.

The General Ledger menu consists of the General Journal, Cash Journal, Posting Update, Financial reports, Transaction file print, Account file maintainence, and disk utilities.

The Account File Maintainance comes with a sample Chart of Accounts for us NOVICES which you want to make a copy of before Initialization to get you familiar with what can be done. A reading of the manual is a must in this area.
1000 accounts may be set up and departmentalized into 99 seperate areas. I only needed 79 accounts for my purposes (real estate investments, incomes, expenses, and a small payroll). The accounts can be modified or deleted anytime and budget figures are entered here also.

The General Journal will handle 500 entries at a time which can be modified or erased at anytime before posting or your next cycle. Screen or printer display anytime.

The Cash Journal also handles 500 entries as above plus you can distribute each entry to a total of nine different General Ledger Accounts. No limit of Cash Account distributions. Screen and printer display anytime.

Posting Update is an automatic printout of everything entered for a cycle from this or the other programs (ar/ap/pr). It's hard to make a mistake, if for example you might have an unregistered account # (Chart of accounts or General Info File) for a transaction, Posting is cancelled till the error is fixed. Transactions are flagged after posting so double posting cannot occur.

Financial Reports will show a Balance Sheet, or Statement of Income & Expenses (month or quarter and year), and a Budget Analysis Report, Zeroing of account. Screen or printer display of course.

Transaction File Print will give you screen or printer display of transactions to date and optional highlited posted transactions.

General Ledger Disk Utilities will zero account file balances, erase Transaction File, merge General Ledger Transactions (more than one disk).

Accounts Receivable program will handle entries, automatically figure taxes, Bill Invoices (preprinted forms available), Invoice Ledger (open, overdue, aging, closed), Accounts Receivable Journal (audit trail), Sales Journal, Finance Charge Update, Print Statements, General Ledger Distribution Accoount File (line items to one of 22 seperate income accounts), Tax Code File (up to nine sales tax rates), Commission Schedule (up to 15 salesmen or other classifications), Customer File (very flexable). This program will hadle 1800 invoices and customers per 8050 disk.

Accounts Payable will handle 2500 invoices and vendors per 8050 disk. Each invoice may be distributed among nine seperate expense accounts. Prints checks (forms available) with up to 13 invoice details per check. Credits and Discounts automatically. Check Register. Aging Reports. Cash Requirements Analysis Report. Full or Partial payments.

Payroll. Maintains current, quarterly, yearly totals for up to 350 employees. Prints checks with full detail. Weekly, bi-weekly, semi monthly, monthly. Regular, overtime, holiday, piecework. All tax plus eight misc. deductions. Prints 941 and W-2's. The Tax File can be updated anytime for new rates (like 1986 Soc. Sec. will be 7.15%). Employees can be departmentalized. Lots of printouts if you want, Files, Transactions, Misc. Payments & Deductions, Deductions, Journal, Check Register, Department Summary, Deduction Register, Absentee Report, Checks (forms available). Check pay for each employee may be entered manually and printed or done automatically. There are 17 programs in Payroll and the ones necessaru for a complete payroll are starred (*), which even here it is hard to make a mistake.

All math in this system is done automatically so the only mistake you make is entering the wrong number.

Unfortunately, as one CBUGer observed there is no inventory program in this system. So here is another challange for our group if it is worth it.

Another asked about multiple and differing discounts on the AR invoices. It appears that they would have to be entered on a one/on one basis, and then the figuring becomes automatic within the program after that. Even here. depending on need, our storehouse of remarkable geniuses could come up with something easier.

The Manual provided is quite descriptive in all areas and sets up a sequence of logical steps as you progress through the system. It repeats all of the prompts displayed in the actual program with simple explanations. It does not seem to go into any possible alternatives or any reasoning behind the various sequences.

Some examples I discovered were - When deleting a Regular account from the Chart of Accounts you must first delete the $ amounts before the DEL command will work. Or filling in paychecks manually will circumvent the required (*) Transaction Entry. Or you really don't need to run the required (*) Payroll Journal for automatic operation though it has to be a good idea with large payrolls.

On the whole, for my purposes (I now have three accounting systems working), I found the system to be trouble free - quite simple - and fast.

Mr. Tingley at CMS is now marketing disks with chart of accounts for Partnership - Corporation - Sole Proprietorship for both CMS and INFO DESIGN.

The CMS ACCOUNTING SYSTEM FOR THE B128 COMPUTER

by: Al Tingley

Many of you B128 owners are not aware of the professional accounting system available from CMS Software Systems of Mesquite, Texas. Some of you who have used the 8032 computer may be familiar with this system under the name Southern Solutions. CMS has been selling this program since 1979 and, in fact, was the first accounting system made available for Commodore computers. Not satified with being first, Chuck Stuart, program author and president of CMS has continued to update, expand, and improve the CMS Accounting System until it has become recognized as one of the finest accounting systems available for any computer. Of most importance is the fact that CMS continues to support the system for both the B128 and 8032 computers. Versions of the same program are current available for the new C128 and IBM and Omega versions are in the works.

The CMS Accounting System III for the B128 is an integrated package of General Ledger, Accounts Payable, Accounts Receivable, and Payroll modules all on one disk. Switching from one module to another is accomplished simply by selecting the desired module from the menu and pressing a corresponding key.

The CMS Accounting System is a fully interactive general accounting system designed by Mr. Stuart especially for the first time user, yet it fulfills the requirements of those advanced enough to want a full purpose office accounting program. The author is an accountant that uses the program to record all the financial activities of 32 businesses of varying degrees of complexity.

Implementation of the program is extremely user friendly. All input requests are fully prompted with complete verification of input data. Important reports can be printed on the screen for user checking prior to printout at any time and both screen and printer reports can be stopped and restarted at any time. The user is led easily thru each function by a series of highlighted prompts fully explaining the required input at each point. A professionally written instruction manual further explains each step and prompt as it is encountered by the user.

THE GENERAL LEDGER will accept up to 1000 accounts in the Chart of Accounts and also allows up to 99 departments. The Cash Receipts and Disbursements Journal allows one check or deposit to be distributed up to 9 separate G/L Accounts. This is invaluable when recording entries such as mortgage payments consisting of principle, interest, and tax escrow entries. Account balances are maintained for Current Month, Quarter-to-Date, and Year-to-Date. A detailed Capital Budget Analysis Report may be printed out at any time to either the screen or the printer. All the modules may be used as stand alone programs or as an integrated package.

THE A/R MODULE allows a total of 1800 invoices and customers per 8050 data disk. Each invoice line item may be distributed to a separate G/L income account. Prints invoices and monthly statements and allows for individualized messages on the statements. Finance charges may be added automatically at a user determined rate and period. Complete invoice aging reports with user set aging breaks may be printed at any time to the screen or printer. An Overdue Invoice Report is also available for close monitoring of receivables. Handles credit memos as well as receivables. A Sales Tax Report is available showing total sales and sales taxes payable for up to 9 separate sales tax rates. Total sales and sales commissions are automatically tracked for up to 15 salesmen.

THE A/P MODULE allows a total of 2500 invoices and vendors per 8050 data disk. Each invoice may be distributed among 9 separate G/L expense accounts. Prints the checks with complete voucher detail for up to 13 invoices per check. Invoices may be paid by vendor range or individual invoice number and partial payments may also be made. Credit memos and term discounts are taken automatically. A Check Register and Aging Report may be printed at any time. A special Cash Requirements Analysis Report is available to assist in determining upcoming cash requirements. Stand alone or integrate-your choice.

THE PAYROLL MODULE maintains Current, QTD, and YTD totals for 350 employees. Prints payroll checks with full deduction and pay details as well as YTD totals. Accomodates weekly, bi-weekly, semi-monthly, or monthly employees. Use regular, overtime, holiday, and piece work hours if desired. Handles federal, state, and up to 8 miscellaneous deductions and payments per employee. Prints out Payroll Journal, Check Register, and an Absentee Report as well as state and federal quarterly tax return data and W-2 forms.

As you can see from the above this is a super program. If you are impressed you're not alone. After using the accounting system purchased with the B128, the CMS system was the answer to an accountants prayer. Since switching over to the CMS system the following improvements have taken place: One program disk instead of four (the G/L module doesn't even require a data disk); screen review before printing; budget analysis and other reports; built in stop print command (invaluable during paper jams); copyable program disks; a master transaction file option (IRS loves this); trailing zeros not needed on account numbers (a real time saver); departmental breakdowns; multiple work station capability; hard disk capability; left (debit) right (credit) entry format; and the ability to go from cash disbursements journal to add a new account and back again without a mandatory printout.

If the above article sounds biased it just might be. The author is the president of Comp/Trac Corporation which distributes the program to B128 owners throughout the United States and Canada. See Comp/Trac advertisement in this issue.

In the old days of manual bookkeeping there was a car called the Packard which boasted the slogan "Ask the Man Who Owns One". This is also appropriate in the case of the CMS Accounting System III.

DIRECTORY RESTORATION

by: Warren Kernaghan

This is a do-it-yourself article about restoring a 'trashed' directory on an 8050 drive with "Diskview", a program on my disk, (CBUG#4). If you are only concerned with a scratched file, use "Unscratch", also on my disk, and save yourself a lot of effort.

This article came about as a result of a friend's disk directory being destroyed while using Superscript. There was nothing recognizable to be found on the directory track and sector (39,1). A look at the BAM (block availability map) told us that there were still files on the disk, so we decided to try salvaging them and learn how in the doing. It CAN BE DONE by those who are not experts. There are probably others not familiar with this, so I'll attempt to go through it understandably.

This is a way to rebuild the essentials of a directory. For an example, two common disk files will be used, first a sequential file, then a program file. Diskview is loaded and the main menu is displayed. Press #1, then input '39,1' when asked for track and sector (T & S).

When a normal directory track (39, sector 1), with only two files, is loaded into the buffer, this is approximately how it would appear:

```
0 255 a&a (38,1) FILENAME1        @@@@@@@@@b@ >2<
@  @  b&b (38,2) FILENAME2        @@@@@@@@@f@ >6<
```

The 0 and 255 indicate 8 or fewer directory entries. If there are more than 8, the next track and sector would appear there, as 39 and 4. The file type and track/sector are encoded (ASCII code) in the 3-character group. The first 'a' is reversed and equates to 129, which means a sequential file. & = 38, and a = 1. The starting track and sector of the files can be seen preceding the filenames, and the number of blocks occupied is shown at the end. Press return to continue.

Choose #2 on the menu and the above block will be converted to an ASCII table display. Stop the display when the numbers in the left hand column reach 64 or so, by pressing return. This is the byte index column for the first directory sector. Byte 0 is just to the right of the 0 in this column. Going across are bytes 1 through 7, then in the next line are bytes 8 through 15.

```
                 First directory entry:  (bytes 2 through 31)
                  0   1   2   3   4   5   6   7
      Index      /   /   /   /   /   /   /   /
        0         0 255 129  38   1  70  73  76   & fil    129=seq, 38=track, 1=sector
        8        69  78  65  77  69  49 160 160   ename1   70 through 49 = filename1
       16       160 160 160 160 160   0   0   0            160 = shifted space fillers
       24         0   0   0   0   0   0   2   0            2 = # of blocks in file
```

Bytes 0-1: 0 = last block used. 255 = next byte to use.
Byte 2: 129 shows this to be a sequential file, and causes the 'seq' in the directory display.
Bytes 3-4: 38 is the first track and 1 is the first sector occupied by this file.
Bytes 5-20: This is the name of the file 'filename1' in ASCII code, plus shifted spaces (ASCII 160) to make up the allotted 16 characters.
Byte 30: This byte displays the number of blocks occupied by the file. In this example it is 2 blocks.

```
   Second entry:
       32   0   0 130  38   2  70  73  76   & fil     130 = prg, 38,2 = t&s
       40  69  78  65  77  69  50 160 160   ename2
       48 160 160 160 160 160   0   0   0
       56   0   0   0   0   0   0   6   0             6 = # of blocks in file
```

The second directory entry is essentially the same as the first, except: Bytes 32-33 preceding the entry are always 0 and 0, unused.
Byte 34 shows ASCII code '130', which indicates a program or 'prg' file.
Byte 62 has an ASCII '6' for a 6 block file.

If more files were added, they would occupy these byte groups:

Entry	Bytes	Entry	Bytes
Entry #3:	bytes 64-95	Entry 6:	bytes 162-191
4:	98-127	7:	194-223
5:	130-159	8:	226-255

If there were more than 8 files on the disk, each sector would have the same format as the first one. Each sector would begin with the track and sector numbers of the next directory entry. Here is the sequence of sectors used:

```
39-1,4,7,10,13,16,19,22,25,28,2,5,8,11,14,17,20,23,26,3,6,9.........
```

What to do if all that directory information is gone to wherever directories go when you can't find them? Well, the disk drive first stores data near the directory track, then spreads it out in both directions. The first file is stored on track 38, sector 1, so the first operation is to trace the blocks starting there.

Press #8 at the main menu and answer with 38,1 when asked for the track and sector desired. We find that the first file in the example occupied blocks 1 and 6 of track 38.

We now know the starting track and sector, also the number of blocks used. What are the minimum requirements for a workable directory? (Considering only sequential and program files). Let's look at the above ASCII tables again. File type, starting track and sector, filename, and the number of blocks occupied; these are needed. For more details, especially on the other less common file types, refer to Commodore's Disk Reference Guide, P.N. 320972-01, chapter 7. (My copy came with the 8050 disk drive.)

If you know the file type, fine, just use '129' for a sequential (text file for Superscript), or '130' for a program file. If you have a printed directory to refer to, it will be a gold mine. Otherwise, select #1 on the main menu and load the first block of the file: 38,1 in this case. This allows you to read the file, and you can see if it is a text file (sequential), or a program. A Superscript text file would look this way:

```
38 6 @ *NB:FILENAME1m @*LM9:rm65mm @cn1m (etc)
```

You can see that the first information is the track and sector for the next block of the file, (where the next part of the file is found), then the file itself is displayed. In this case, the name of the file is apparent, but even if not, you can guess by the content. This data is easily recognizable as a Superscript text file, which would be sequential. The small 'm's are returns.

A program file also begins with the next track and sector in bytes 0 and 1, and has up to 254 bytes of the program in CBM memory format, with key words tokenized. BASIC token lists are not easily found, although 'The Complete Commodore Inner Space Anthology' published by Transactor magazine has a SuperChart that includes BASIC tokens, ASCII, screen characters, and others. For instance: REM will appear as a reverse 'o' when you look at a program block with Diskview. PRINT will be a reverse 'y', READ is a reverse 'g', etc. The remarks and print statements can be easily read, as they're not encoded.

We now have all we need to restore the first directory entry. We'll load the directory block, convert it to ASCII display, then move to the modification mode. Press #1 at the main menu to reload the directory T & S, (39,1), then return to the menu, press #2 to display the ASCII table, and stop the display when it reaches byte 40 or so by pressing return. Return to the menu and press #3 to restore the directory entry. When the block is displayed, stop it at about byte 40 again. Now begins the actual restoration.

"Which 'i' do you wish to modify?" is displayed at the bottom. The program is asking for the byte number, so if your original directory had 8 or fewer entries, bytes 0 and 1 should have 0 and 255 displayed. If more than 8 entries, 39 and 4 should appear. Look along the row to the right of the 0 in the first column, in the first two spaces.

To change byte 0, type 0 in answer and press return. The ASCII of the code present in byte 0 will be shown, and you are asked for the new value. Enter 0 or 39 as appropriate. When this is done, the question repeats, so if byte 1 is not correct, enter 1, return, then the correct ASCII code (255 or 4) for that byte. You are not changing the disk yet, so don't worry about what you are doing. Practice on a disk with files you don't care about, or copy one of your other disks to play around on.

Byte 2 is the file type, so enter 129 if sequential, or 130 if a program file, if neither number appears there. Bytes 3-4 will always be 38 and 1, so enter those if needed. Bytes 5-20 are the name, entered in ASCII code. I suggest using the letter 'a', ASCII code 65,' for the first file, 'b', ASCII 66, for the second, etc, then renaming when finished. The alternative is to convert the original name to ASCII ,then entering it byte by byte. Take your choice. If the name is shorter than 16 characters, fill in the rest with '160's, which is ASCII code for shifted spaces, as shown in the example ASCII tables. This will take you through byte 20.

Zeros should appear in bytes 21-29. Byte 30 should show the blocks occupied by the file, and byte 31 is 0. That finishes the changes for the first directory entry. Press return, and the display will show the changes made. Be ready to stop by pressing return. At the bottom, 'OK?' appears. If you press 'n', you are asked again which 'i' or byte to change. A 'y' will give you a last chance to return to the menu before starting the disk writing sequence, cancelling the changes. If you press 'y', you will have to start over. You normally press return, choosing the default 'n', and are asked the destination T & S, with the current T & S displayed. For the first directory entry, this is 39,1. The last query to answer before writing to the disk is the destination drive. Don't type 8 or 9, just 0 or 1. The data is transferred to the drive buffer, then written to the disk.

You can check your work by pressing #7 on the main menu, then #4 to display the directory. For the entry corrected, the blocks used, name, and type of file should be as you wanted them to appear. If not, go back and make corrections, starting with #1 on the main menu, etc.

The second directory entry is done the same way, except that the first two bytes are 0 and 0. The second file entry of the example starts on track 38, sector 2, so press #8 on the main menu to trace the blocks used and learn the number of blocks occupied, as we did before, then follow the same procedure.

The detective work gets a bit trying after locating the first file or two. The next file beginning depends upon where the previous files left off. Make a grid table if you have quite a few files, and keep track of which blocks have been used, and preferably for which files - a, b, c, etc. There are 77 tracks with 29 sectors on tracks 1-39, 27 on tracks 40-53, 25 on 54-64, and 23 on 65-77. Mark the unused blocks by displaying the BAM (#6 on the menu). This will save you much effort in viewing blocks that are not used.

You will notice a pattern as this is filled in. As you run a trace on a file which gives you several blocks, and find that the first block you have isn't the beginning of the file, you can usually tell which blocks are the most likely to be closer to the start of the file. (Blocks and sectors are used interchangeably here).

For instance, if you ran a trace starting at 38,12 and found sectors 12, 17, 22, and 27 used, but find that 38,12 isn't the beginning when you load the block for display, its time to check the pattern. Note the spacing between the blocks? It is in multiples of 5, so the next place to look, toward the beginning, is backwards in multiples of 5. First would be 38,7, then 38,2. The normal step is 5, except when the beginning of the sector is reached.

Here's how a sample 143 block file progresses:

```
35-1,6,11,16,21,26,2,7,12,17,22,27,3,8,13,18,23,28,4,9,14,19,24,20,25
34-0,5,10,15,20,25,1,6,11,16,21,26,2,7,12,17,22,27,3,8,13,18.23,28,4,9,14,19,24
33-0,5,10,15,20,25,1,6,11,16,21,26,2,7,12,17,22,27,3,8,13,18,23,28,4,9,14,19,24
32-0,5,10,15,20,25,1,6,11,16,21,26,2,7,12,17,22,27,3,8,13,18,23,28,4,9,14,19,24
31-0,5,10,15,20,25,1,6,11,16,21,26,2,7,12,17,22,27,3,8,13,18,23,28,4,9,14,19,24
30-0,5 (end)
```

As file beginnings are found, keep a table of the file type (seq or prg), starting T & S, filename, and the number of blocks occupied. That's all you need. When all occupied blocks are accounted for, or you decide its not worth it, or all major files to be saved are found, go through the procedure to finish the directory entries. The hard part is over! Just rename any abbreviated filenames. You may have to load the files to decide what the names should be. Files loaded by other programs require the exact name that the loading program is looking for.

Yes, it IS a lot of work, but sometimes it's worthwhile. Have you been making backup copies?

8050 SPEED REGULATION

from a letter by: Warren Kernaghan

It appears you are correct about disk rpm error causing Superspell problems. You'll recall my recent letter describing the loss of an edited file if the "superspell.tm" file was not scratched.

The Physical Exam showed the rpm on my 8050 drive 1 to be about 300 rpm when first starting, then gradually slowing to about 292 rpm when warm. At 80 degrees ambient temperature, this took only an hour or so; longer when cold weather set in.

In warm weather, it was no real problem to set the rpm to about 299 when warm, and just wait until the rpm stabilized when I wanted to copy programs, etc. In cool weather, the wait was too long, so I decided to try tracing enough of the rpm control circuit to do some trouble-shooting, thinking the analog circuit couldn't be all that complicated.

It did take awhile, but was worthwhile in gaining familiarization with the drive. The problem was found and cured. A 6.8 volt zener diode (CR21) that regulates the voltage applied across the rpm adjusting potentiometer (R50) was leaky, resulting in poor regulation as the surrounding temperature changed. Replacing the original 1N5235 diode with a more commonly available ECG 5014A solved the problem. Total change from cold to warm is now about 1.5 rpm in 4 hours, and there have been no problems with Superspell since then.

The potentiometer output voltage is connected to one of the inputs of a 741 operational amplifier (U12) used as a comparator. Tachometer voltage from the drive motor is rectified and feeds the other opamp input. The 741 output then goes to a darlington power transistor (Q1) which controls the DC voltage to the drive motor.

<<Component locations:

Drive #	Zener	Pot	Bias Resistor	Output Transistor
0	CR25	R61	R59	Q2
1	CR21	R50	R49	Q1

NOTE: This information applies to all but Tandon Drives. Tandon drives have their analogue board laid accross both drives left to right (though the circuit schematic is essentially identical). Few 8050 drives use the Tandon design, so all the rest have their analogue board mounted exclusively atop drive 0, which is what all of this information relates to. Warren suggests that the zener be replaced by a ECG 5014A, which we have in stock in quantity -- see the order form. HOWEVER, a better choice is the LM329 precision reference diode but you have to replace the Bias Resistor with one of 2.7K value as well -- a slight bit more work. We also have the LM329 and 2.7K resistors available thru the library order form.

A word of caution. Do not attempt this surgery unless you are or have help from a competent electronics technician. He need not know computers, but must be experienced in circuit board soldering/repair techniques.

A second caution. There are other causes of speed errors, however since the diode problem is so severe and pervasive it is manditory to correct it first.

Of course, you will have to have a method of resetting speed once you have done the surgery. You must therefore load physical exam into your computer before surgery, or do surgery on only one drive servo at a time. If you choose to load into the computer, make sure that your power cords are properly grounded, then, after loading Physical exam, disconect the IEEE-488 cable while the power is turned on; turn off the drive ONLY, perform the surgery, and reconnect. Unlike most other data lines, the IEEE can be disconnected during operation provided proper grounding remains at all times during the connection/disconnection. This is because the IEEE contains only ground lines and buffered signal lines -- no power or B+ of any kind.>>

8050 SPEED -- A COMPREHENSIVE SOLUTION

by: Gerry Mills

This note reports the development of a modification to the 8050 disk drive to eliminate the serious speed errors that occur in the system. These errors plague the user because as designed the 8050 did not correct for the drift of the tach sensor itself! Efforts to correct the poor stability of the reference voltage and the overly sensitive and unstable adjustment 'pot' often fail because the tach does not supply correct VOLTAGE data!

The good news is that the tach can be used so that the FREQUENCY of the tach rather than the gnerated voltage is used to control the drive. This is done by comparing the frequency output of the motor-tach to a crystal oscillator frequency. Therefore absolute non-drifting speed accuracy. We have included a provision for fine tuning the speed; the test disk

PYSICAL EXAM is ideal for this.

This has been done successfully and we can report that the speed can be made to stay put even from a cold start! No longer necessary to run for three hours just to get the drift to the point where it will stay put for 15 or 20 minutes after resetting!

This mod will be offered as a service to the CBUG members within a few weeks. The design is now being put into a printed circuit format that can be installed on the analog board of the 8050 (the pc board that is mounted on top of the drive 0 frame). We will offer to do this in any of three ways: **(1)for those who can remove this board and send it to us; **(2) as a kit for those who are SKILLED ENOUGH OR HAVE ACCESS TO SOMEONE WHO IS ---don't try it if not really equiped and familiar with computer electronic as the well meaning efforts of anyone not really skilled and equipped can seriously damage this necessary part of your 8050 -- and repair of this board would not be undertaken by us or probably any competent outfit for less than $150!!!;**(3) by shipping to us the PROPERLY packed entire 8050 unit.

At the present we cannot give a firm price but it looks to be quite reasonable for the pc construction and if we receive enough interest the job can be run in our service department on a routine basis---it would of course be least expensive for the option 1 approach -sending in the analog board- and perhaps a local technician could even remove it and then reinstall the upgraded board upon its return. The 8050 will not operate without the analog board so we would endevor to turnaround the boards received in two or three days and can ship UPS surface or any other way requested at additional charge. Whatever the cost turns out to be we will be contributing ten percent to the CBUG to support the clubs efforts to improve and expand the utility of our systems.. The next news will give the details.

G.B. Mills, c/o SEI SERVICE CENTER
P.O. Box40, Hillsborough NC 27278-0040
(919)732-6131 -- or best call or write Norman expressing your interest.

<<All of this tech talk about drives must catch most members by surprise. What this really says is that most people have not experienced the problems with sufficient severity that they become noticable. The unfortunate members whose drives really act up are however in serious trouble. Most importantly, don't take a complacent attitude as to the problem never striking you. Minor data errors can occur, excessive re-read and re-write attempts by the very smart 8050 drives may be masking the problem or very likely you have a drive that is not substantially afflicted. BUT, it is well worth finding out and being on top of the situation. The B128 & 8050 system is generally an exceptionally stable system -- infact the only major flaw known to date is the drive speed error problem! Whip it before it catches you when you can't afford to go down!!>>

MEMORY EXPANSION OPTIONS IN THE B-128

by Gary L. Anderson
1528 34th St. SE.
Cedar Rapids, Ia. 52403

I would like to explain some options with regards to increasing memory in the B-128. Most of these options I have implemented in by B-128 and would like to report to the CBUG membership my configuration. Most of us are aware of the unused banks namely 0,5-14 but there is also a wealth of space in bank 15 which is presently going begging. The following paragraphs will discuss both areas and ways of implementing them. But first I must make a plea that if you get a brain storm and start work on something... anything... hardware or software... please register it with the club project registrar Marilyn Gardner. I have received about a half dozen phone calls from people interested in memory expansion only because I have registered my projects with CBUG. Lets communicate and plan for progress. As I will get into later, there are different ways of implementing 1 Megabyte in the B-128 and if we don't all use the same addressing scheme then compatibility with a future co-processor could be a very sad learning experience for those that use a "non accepted by the majority" technique.

BANK 15

This is where the operating system lives and is a very easy place to execute machine code, no fancy transfer routines, and just a SYS gets one there. It turns out that there is more address space available to the user in bank 15 than there is used up by the operating system! $0400 to $7FFF and $C000 to $CFFF is there for the taking, almost 37K of memory.

BELOW THE CARTRIDGE: 0400 to 07FF is already on the main pc board as the top half of a 2K ram IC and appears not to be used by the operating system. $0800 to $0FFF is a 2K block that is decoded in hardware as EXT BUF CS and is used by the co-processor schematics that I have seen. $1000 to $1FFF is a 4K block that is decoded in hardware as DISK ROM CS. I am implementing EXT BUF CS and DISK ROM CS, a 6K area, on my 1 Megabyte expansion circuit board by using a socket for an 8K ram or rom IC with the hope that some machine language software genious out there will write a ram disk utility. I am providing the capability to de-select $0800 to $0FFF, EXT BUF CS, on my 1 Meg board if co-processors should ever come on the scene since this 2K block will also be present on the co-processor board.

THE CARTRIDGE: $2000 to $7FFF is reserved for the cartridge port, a 24K area, and I have implemented this area with my 24K RAM CARTRIDGE CIRCUIT BOARD. (see ad this issue). I presently am running BMON by Larry Scott in my cartridge. BMON is a machine language monitor that quite frankly leaves Micromon of the PET in a cloud of dust. Larry has sent his disk of BMON to CBUG for release to the membership. If you are into machine language don't pass this one up. The BMON disk comes with complete instructions in Superscript II files.

ABOVE THE CARTRIDGE: This is the only area I have not touched. $C000 to $CFFF is a 4K block and no decodes are available on the pin fields inside the B-128. Two ICs would be needed to decode, one to determine bank 15 and the other to determine $CXXX. There is another small area from $D900 to $D9FF, 256 bytes, that is decoded by the line CS1 on the pin fields P6 and P7. It looks like an ideal area for a special peripheral IC.

BANKS 0 THROUGH 14

If you think bank 15 sounded interesting..... hold on to your cursor! Industry jargon would call this area 960K bytes, 15*64K, actually its 983,040 bytes. The reason that one hears the term 1 megabyte for this area is that it is implemented with 256K by 1 DRAM ICs, a total of 32 of them. The operating system resides in bank 15 therefore the 6509 can't get to the DRAM that is actually there in bank 15. The schematics of the 8088/8087 co-processor however show that the co-processor can access the bank 15 DRAM, making bank 15 a nice little 64K hideaway for the co-processor. Conversely the co-processor can't get to bank 0 of DRAM, the co-processor addresses bank 0 for its own operating system and I/O, making bank 0 a nice little 64K hideaway for the 6509.

IMPLEMENTING 1 MEGABYTE: Upon originally looking at the schematics of the B-128 I pondered what it would take to implement 1 megabyte. I decided that modifying the main board was an option but it appeared too labor intensive to do on a repetitive basis, besides I figured that many B-128 owners would not want to tackle that kind of sensitive desoldering/soldering job on their static sensitive main circuit board. I began looking at the possibility of a drop-in circuit board since the pin fields P6, P7, and P9 of the main circuit board appeared to have the necessary signals. This would make the upgrade to 1 meg a simple 5 minute job. I discovered that there was enough room on the expansion board for an 8K socket for the unused address space below the cartridge in bank 15 from $0800 to $1FFF, the perfect place for a ram disk utility. I also determined that there was space for pin fields on the 1 meg expansion board. This would be important to provide the option for a future drop-in co-processor. The pin fields will be extensions of P6 and P9 and will have an identical mechanical foot print to the main board's P6 and P9 allowing a co-processor board to fit on either the main board or the 1 meg expansion board. I did an anlysis of +5V current consumption and the results were very encouraging. My stock B-128 draws only 2.52 amps from the +5V line and with the addition of my hand wired 1 meg prototype board that increased to 3.13 amps. The power supply is rated to deliver 5.5 amps on the +5V line so this leaves pleanty of margin for a future drop-in co-processor. An added feature of increasing the load on the switching power supply is reduced heat disapation from the supply. The 115 VAC input current will actually drop as the +5V current is increased from its present low value indicating more efficient operation.

DEFINING A8: The 256K by 1 DRAM use another address line in comparison to the 64K by 1 chips presently on the main board. Consequently another multiplexer must be added to the main board or 1 meg expansion board to control the row, column, and co-processor addresses. I hereby would like to put forth a definition to be adopted by the membership of CBUG. I am defining pin 33 of P9 as "EXT MA8". This pin is previously unused. When the co-processor takes control it forces the main board or 1 meg expansion board address multiplexers to accept the address on the EXT MA0-8 lines only! The co-processor then multiplexes row and column addresses on the co-processor board and sends them over the EXT MA0-8 lines. Without the addition of the EXT MA8 line the co-processor can't access 1 meg of memory. The co-processor schematic will also need a slight modification. Another 74LS257 must be added to multiplex PB0 and PB1 as the row and column addresses respectively. If you feel that I am off the wall with this definition write me and present your case, this needs to get settled at the front end of what I expect to be a 1 meg memory explosion among the membership.

RUNNING 1 MEGABYTE OF MEMORY: What happens when the power switch is flipped? The operating system of the B-128, or should I say B-1MEG, goes off to see how much DRAM is present, and takes about 27 seconds to return the standard opening prompt. PEEK(855) or PEEK(861) will tell the bank number that is top of DRAM memory. Superscript II presently will allow the user access to banks 2 through 9 with an ESC J, 512K of available memory. The Superscript II manual on page 84 claims that you can run banks 2 through 15, this must be a misprint. Who would like to fix that and provide access through bank 14? The 8432 Emulator is amazing! Imagine... 15 8032s in a B-1MEG. When running the 8432 Emulator don't forget to modify the basic loader. In lines 120 and 130 REM out the FRE(X) statement just before the FRE(X). I'm still analyzing to see how well the emulator works with that many 8032s defined.

IN CONCLUSION: There is a lot of address space in the B-128 that can be put to excellent use. Some hardware implementations already exist, others are in development to be released soon. Some existing software will already make use of the expansion, either automatically or with minor modification. It should be interesting to see what new software comes on the scene to make use of the incredible expanse of 960K bytes. Lastly, those interested in developing a ram disk utility should register your intentions with the project registrar. Hopefully you can all get together and define a good set of specifications and a uniform command set.

Note: Mr. Anderson is a SR. Engineering Technician with Rockwell International and has published articles with both THE TRANSACTOR and '73 MAGAZINE. He holds an advanced class amateur radio license as WB0GWP.

AND FROM THE WILD WILD WEST

several articles by: Joe Rotello

ON SUPERSCRIPT

Superscript II (SSII from here on in) is quite obviously a masterpiece of programming. It even has features that STILL have not appeared in other micro based Word Processing Packages even after three years. A few things that may be obvious to both new and experienced users, or may not be, are as follows:

a) The SSII master or original disk is well copy protected. Using a good disk copy program like Knights, we find that the COPIED disk will exhibit ALL the "safeguards" as the original disk did. This means that the original/copied disk will not allow any file to be re-written to the disk. In turn this means that you may not upgrade any file on either disk and if you try, the 8050 DOS will generally reply with a dreaded "73" (standard DOS protection) or the disk will spin, appear to update the file, but the error LED light will turn to red and the update process will be halted and not performed.

b) Elizabeth Deal has worked up a neat "patch" program ("pre-superscript", on the UTILITY BD, 6-85 disk, among others). A worthy program to use but note a few items:

- It will "fix" the erratic cursor problem many versions of SSII have but will do so at the expense of a slight loss of cursor speed, most noticeable when the cursor is driving across a long line horizontally. As the cursor nears the end half of the line, it begins to slow down.
- The "pre-superscript" patch must be saved to a disk other than a SSI master/original or a copied master disk (master backup) for reasons made clear in (a) above.
- When using the "pre-superscript" patch, note that the program loads in "superscript*" as the "load and run file". If you need the special "ss2" printer setup tables for your particular installation, change the program load line to read

"ss2". This will load in the printer tables (generally from a disk in drive #1), and will then proceed to load in the main SSII program as usual. Its been noted here that most versions of SSII will function properly with most letter quality printers without the user HAVING to load in the ss2 printer tables. But there may be times where you have set up a different or special printer and the data above regarding ss2 tables may be important.

- You can use SSII to save SEQuential text files to disk, especially if you are using a modem or running a BBS system, but the SSII manual does not make this "feature" obvious enough. At the OUTPUT function, select "S" and you can save the text, AS IT WOULD APPEAR ON A PRINTER to the disk as a SEQ type text file. This infers that if you have the text formatted (margins, form feeds, etc) you best remove them (i.e, leave the text as unformatted as you can) or the BBS (or whatever system uses your text file) will follow all those formatting features, producing oddball looking text or text that has 25 line feeds where you dont want them!!

- -

CAN WE COMMUNICATE?....

(c) 1986 Joseph Rotello, 4734 East 26th St,Tucson AZ 85711

Aside from being a fine representative of quality micro engineering, the B/700 series from Commodore excels as a tool for telecommunications. Teamed with a MODEM and a worthy telecommunications package, the B/700 becomes an effective tool for use with other computers, databases, Bulletin Board Systems and professional data systems such as WestLaw(tm), Dow Jcnes News Retrieval and many others.

Strangely enough, the B/700 has had available terminal software nearly from the beginning but the software was generally sparse as far as user features went. Likewise, B/700 compatible modems have always been around, but the best modems still need good modem or terminal software to perform effectively.

Enter TWO very good B/700 peripherals, the BeeLine(tm) Terminal Software Package from LemData Software, and the new-version Password 1200 Modem from U.S. Robotics. By themselves, not at all an effective product...together, a very powerful and easy-to-use system in a most sensible price range.

We begin by taking a short look at the Terminal Software and then tying them together to see how they perform in tandem.

Any modem or terminal software package can turn a computer into a data input/output machine. But it takes a COMPLETE software package to allow the end-user to perform meaningful functions with their computer and BeeLine falls into the second category. The package is very un-pretentious - in fact, even the user manual is stored on the program disk and one gets an odd feeling when you receive the software package and it all fits in your hand.

Once BeeLine is loaded and run, all fears vaporize. This is a WELL done piece of software that is genuinely effective and simple to use.

MENU, PLEASE.....

Basic BeeLine operations are carried out by two main Menus each easily selected and well documented ON THE SCREEN ITSELF. Moving from menu to menu is a breeze - a credit to LemData that they too saw that a powerful terminal package with many options must be able to display those options in an easily understood, uncluttered manner.

The first menu controls such options as the built-in text buffers, saving and loading the buffer contents to and from the disk, setting up the receive file, transmitting a disk file to a remote system and more. A second menu allows the user to view disk directory, create or modify their own Setup File, change function key designations, modify the built in ASCII/PET conversion tables, select from various baud rates and change other operating parameters such as parity and echo.

THANKS FOR THE (LARGE) MEMORY...

One nice touch is that BeeLine takes full advantage of the vast memory available in the B/700 units and of the B/700 Functions Keys as well. For example, from Menu 1, the user has only to press Function Key #1 and voila! the user manual is automatically loaded and displayed, without disturbing the program in any respect.

Depending on the B/700 installed memory, two (B128) or more (B256 or above) LARGE memory buffers can be used to store data from the remote computer, files from your disk drive or any number of other data sources, and again, working with the buffers is quite simple as they have their own readily understood sub-Menu allowing full user control. Among the nice finishing touches, in terminal mode the user can turn the buffer on or off from the keyboard with a single keystroke. The user has two or more buffers from which to choose and it is quite possible to load one buffer with data being received from the remote computer <<download>>, switch to the other buffer(s) and upload that buffer's contents to the remote computer.

BeeLine also offers complete compatibility with just about all "Smart Modems" that use the established AT command set. This allows the user to perform such maneuvers as auto dial, turning the modem monitor speaker on or off (if the modem is so equipped) and to monitor the modems progress or any text messages that may be sent by the modem itself to the computer in terminal mode.

One very welcome idea is the ability to fully control the modem data parameters including the various baud rates, parity, word size and echo very simply. While in a Menu, the user merely hits the number key corresponding to the required parameter and, by striking the same key repeatedly, all the various possibilities associated with the parameter are paraded to the user. The user merely has to "stop the parade" at the required parameter and that is that. In short, setting up the various parameters, or changing them, has never been easier.

In addition to providing operation at all of the standard baud rates, including 300, 1200, 2400 and others, the user has full options to change just about all of the BeeLine parameters, telephone numbers and default values, save these values to disk and they will be automatically recalled the next time the program is used. This "remember it-save it-load it again" feature alone saves many minutes of user interaction that might otherwise be required each time BeeLine is run.

Last but not least, the BeeLine User Manual is quite effectively written. The manual is formatted in a full 80 column manner and this not only makes for easy reading, but also allows the manual to fully cover all aspects of the BeeLine program. In addition, a printer hardcopy of the manual can be easily made at any time for ready reference.

All in all, the BeeLine Terminal Program is a great step forward and makes your B/700 system even more of a powerhouse and, at a suggested list price of under $50.00, BeeLine is equal to or better than packages for other micros selling at double or more the price, and, equally important, will be updated by LemData from time to time with worthy or important user features or enhancements. The BeeLine Terminal Package is available from various sources as listed below. Definitely a recommended purchase.

Further information concerning BeeLine Terminal Software is available through:

CompuSystems Management (CSM)
4734 East 26th Street
Tucson AZ 85711
(602) 790-6333

CBUG (National B User Group)
4102 North Odell
Norridge IL 60634

Producer:
LemData Software
P.O. Box 175
Dover PA 17315

- -

ON 8050 WRITE PROTECT ERRORS

Some 8050 MPI drives have a habit of calling WRITE PROTECT errors when you are HEADERing a disk, COPYing to the disk and so forth and you know darn well that nothing is wrong.

```
**************************************************************
* WARNING! DONT OPEN YOUR 8050 IF YOU ARE NOT WILLING TO     *
* PERFORM THE FIXES NOTED BELOW!! IF YOU ARE NOT TECHNICALLY *
* INCLINED, LET A QUALIFIED TECHNICIAN DO IT!                *
**************************************************************
```

The most frequent cause of these errors is due to the WRITE PROTECT SENSORS located at the left front of each drive, just inside of the drive itself. Both the upper and lower sensor halves usually have a sky-blue two lead connector.

Because of the sensor location, the offending drive SHOULD be removed from the housing (a MUST if the WP problem is in drive #0). If the problem is in Drive 1, you dont HAVE to remove the drive (on drive 1, you can carefully remove the left side plastic plate holding the sensors), but it sure helps! <<With Tandon Drives, the drives must also be removed>>.

The wire leads on the upper and lower half of each sensor are silver coated. These leads can get very oxidized especially if the unit is used in a dusty or "grimy" atmosphere. When they get oxidized (sort of a black or dirty dark brown appearance) the the WP sensor connectors can not make good contact at all. Hence, when that happens, the drive thinks that the floppy disk has a write-protect tab on and refuses to header, copy or save data.

In order to inspect the sensor leads themselves, you have to GENTLY pull off each connector and shine a small light on the now uncovered leads. If they are dark in color, very unshiny or otherwise dirty, there are various ways to clean them SAFELY but one in particular works just fine:

```
**************************************************************
* WARNING! THE SENSOR LEADS CAN BE RATHER EASILY BROKEN OFF  *
* IF YOU BEND THEM TOO MUCH! THESE MPI SENSOR ASSEMBLIES ARE *
* VERY HARD TO FIND SO TREAT THEM LIKE GOLD!                 *
**************************************************************
```

GENTLY rub each sensor lead with a SMALL pencil eraser (not a pen eraser, its too harsh!) until the lead is shiny and/or the pencil eraser shows signs of darkening (the darkening is the oxide from the sensor lead and is a good sign that the sensor lead is clean/becoming clean). If you cant reach BOTH sides of the sensor lead (and usually thats the case), DON'T bend the leads to reach them! Just do the best job on the sides you can reach.

After cleaning the leads, moisten a small Q-Tip swab with 91% (or better) Isopropyl Alcohol and GENTLY clean off any residue left on the sensor leads, both top and bottom sensors. Use a VERY pure alcohol since we want no residue left on the contacts - NEVER use ANY other solvent other than TCE (common in "Tuner Cleaners") which should be used by professionals anyway!.

GENTLY reassemble the sensor lead contacts to the sensors and (if it was pulled) re-install the drive(s)

<<It is also reported that there have been numerous defective 7407 chips. This is a 14 pin Integrated Circuit located on the small board on the rear of drive 1. It serves as a buffer (amplifier) for the write detectors in both drives. In an emergency you can simply short the detector leads or short the output pins of the offending gate of the 7407 to ground (pin 7 of the chip). The 8050 will then never see a write protect tab. 7407 pinouts are:

drive#	input pin#	output pin#	
0	3	4	Ground = pin 7
1	1	2	+5 = pin 14

Note that this information applies only to the majority of drives having an analogue board mounted entirely to drive 0. Some Tandon drives have an analogue board extending the full width of the machine stradling both drive 0 and 1.

NEVER tamper with the speed adjustments or do anything that might alter the speed characteristics of your drive unless you have Physical Exam or some other method of resetting speed!!! This may be presumed to include all but the most gentle handling of the drives and boards when dis-assembling a 8050.>>

- - - - - - - - - - - - - - - - - - - -

ON THE SPEED UP PROGRAM

The "speed up 8050" program is an interesting exercise in changing the thinking of the original 8050 dos programmers. The program essentially jumps over now useless internal motor and other routines in all versions of the 8050 including all dos 2.5 roms (-03, -04 (or) -06, -07), all dos 2.7 roms (-88701, -88801) AND all makes of the 8050 (Micropolis, MPI or Tandon).

The effects of SU (as I will refer to it herein) are as follows:

a) The program can be loaded in as a "stand alone" utility or merged in a basic program. For conventions sake, if I merge it with a basic program, I make sure that the execution of the SU code is the first thing that takes place.

b) The code is merely loaded into the 8050 itself. The code DOES NOT have ANY adverse effect on ANY of your programs and I have observed no adverse effect on Knights Copy Utility either.

c) The code is "activated" on the next disk access following the loading of the code into the 8050 and stays active until (1) the 8050 is reset or turned off (or) (2) the B128 causes a disk reset. It is somewhat possible that an exotic 8050 dos program could overwrite the SU code in the 8050 dos, but I consider that rather remote.

d) The code forces the disk access to be near immediate each time the drive (either drive) is accessed and, after each disk access, the drive(s) will continue to spin for about 15 secs or so. This effect is noted in all 8050's and happens to be most noticeable in the Micropolis 8050's that have the screw type head stepper system.

As an example, the 8050's will now activate, access and load programs MUCH faster than before the SU code was installed, I noted that that on a rather long directory, my 8050 MPI (without SU code) took a total of nearly three secs to activate, find and load a moderate size program. With SU code, the same process took about 1.5 sec or so. Note that the code seemingly does not increase the actual LOADING speed, but the speed up effect is none-the-less impressive.

<<The spead up program is located on CBUG Utilities #2 as program name "q", and appears in print in the fall issue of THE CBUG ESCAPE>>

- - - - - - - - - - - - - - - - - - - -

NOTES ON CHANGE LOAD ADDRESS V1.2

The CHG LOAD ADDR V1.2 program present on NORMS' UTILITY COMPOSITE (dated 02 Jul 85) and on other CBUG disks as well, is a "very good idea" type of program. Except for one slight oversight, it will perform well, albeit a trifle slow, and its a dandy for allowing programs made on one Commodore computer (say a B128) to be loaded into and worked on by another computer (say a CBM 8032).

The slight oversight concerns a variable used in four lines in the original program. This variable "el" happens to be a protected variable in versions of CBM basic used by one or more of the other Commodore computers (including the B series!)

The fix, thusly, is to load in the CHG LOAD ADDR V1.2 program and replace ALL occurances of the offending variable (el) with another, less destructive one. The lines to change "el" in are:

```
57 eh=h2+int(c/256):el=l2+c-256*int(c/256)
58 if el>255 then el=el-256:eh=eh+1
81 printmid$(x$,el/16+1,1);
82 printmid$(x$,fnm(el)+1,1)
```

Check your changes carefully and don't confuse "el" ("e ell") with e1 as both variable notations do look alike at times.

I suggest changing "el" to "ex" or something like that - anything that will NOT be a protected variable. Don't touch "eh", as it's pretty important to the program.

Having thusly saved the program from an untimely death, re-save it to the disk...I suggest saving it under a NEW VERSION #, "CHG LOAD ADDR V1.3" for example, so as to set it apart from the "old" version.

Although this "idea" may be obvious, those who have access to Petspeed for the C-64 or for the 8032 will find CHG LOAD ADDR V1.3 speeded up greatly (about 5 X faster overall) and if you have the means to compile it, do so. Even though the Petspeed version of the program will not run directly on the B series, you can still use your trusty 8032 to change the load address for any other Commodore computer.

If you have any questions or comments, you may reach me:

CSM/Joe Rotello Jr.
4734 East 26th Street
Tucson AZ 85711
(602) 790-6333 (Please call at reasonable hours only)

BASIC SCREEN FORMS

by: Allyn Uptain

Synopsis: Simple screen form handler in Basic for data file maintnenace
This file is an explanation of the program "screenform"

<<Note: this is the documentation file for a program "screenform" contributed to the CBUG library by Mr. Uptain and available on one of our library disks>>

B128 Basic provides an easy to access and easy to use screen handler. You can use the "input" statement instead of the "get" and still have control of the cursor with a lot less programming. What is this well kept secret command? It is just the screen defining memory locations 220, 221, 222 and 223. (hex 00dc through 00df). By "poking" these locations, you can define the useable portion of the screen. As you process the fields on a screen, you simply poke the values needed to limit the screen size and location to the one field you want to be "input". This works very well. If you place a default value on the screen before the "input" command and the operator presses return without keying any values, then the default will be returned as if it had been keyed. The insert/delete key is fully functional without moving the rest of the screen. You can modify a default or prior field value without rekeying the whole field and without making your program overly complicated. Let me explain in more detail -

The Bare Facts:

```
poke 220,n     sets the top line of the useable screen.  0 <= n <= 24,
               which are the 25 lines of the screen, numbered 0 thru 24.
poke 221,n     sets the bottom line of the useable screen.  0 thru 24.
poke 222,n     sets the left margin of the useable screen.  0 thru 79,
               which are the 80 columns of the screen.
poke 223,n     sets the right margin of the screen.         0 thru 79.
```

In order to define a one field window for input, then we need to poke the line number twice (220, 221) and the left and right margins of the field (222, 223). experience will teach you, as it did me, that you must poke 222,(left margin of the field minus 2) to allow for the "? " generated by the "input" command and poke 223,(right margin of the field plus 1) to allow room to type a full field and still be able to hit the "return" key.

Setting It Up:

Of course you want to use subroutines to handle your input and therefore do not want to code each set of pokes for each input field. The easiest way to do that is to set up a data table of your field locations and then have your program walk through the table as it accepts the keyed data. The data table must have values for each field to be processed and must include the following:

1. line # of field (0 - 24)
2. starting column minus 2 (0 - 79)
3. ending column plus 1

In addition to those three you will probably have a need for a field attribute indicator. This is just a code of some kind designed to tell your print commands what kind of a field you are processing, i.e., alpha, integer or real. In the following example program we will use the 4th data item to identify the data type where 0=alpha, 1=integer and 2=real.

The Program:

Lines 10-20 set up "F1" for tabbing forward, "F2" for tabbing backward,
"F7" for end-of-job and "F10" for right justifying a field.
Lines 120-123 is the data table that will load the field attribute array
Lines 200-320 setup and display a screen "form".
Lines 340-540 process the input fields and check for data entry errors.
Lines 5010-5290 is the input subroutine.
Lines 5910-5990 is an error message subroutine that either displays an
error message or blanks the message line.

The arrays d$, d% and d are the screen default values. Once a field on the screen has been accepted into the program its default is loaded with the value of "an$" - the input field. If you use F1 or F2 to tab out of a field, then the default value before you entered into that field will be redisplayed before continuing on to the next field. This is nice when you see you have messed up a field, you can just tab out to restore its default. When changing records stored on disk, at line 360 you would read in your disk record, set the corresponding default array elements to the value of your disk record and presto you're ready to display all the defaults. At line 5140 you can add another function key check to tell you when to take the keyed input and write it back to disk.

The fields lo% and hi% control which fields will be processed. By changing these you can "protect" certain fields from the input subroutine. Alternately, you can build a table of protected fields and check that table at line 5050.

The most inconvenient aspect of this type of screen handler is having to hit F10 to right justify the fields. Esc "q" does work to clear to the end of the field, but that too requires a lot of keystrokes. In actual use though this is not as much of a problem as it is when playing with the sample program.

One warning. A syntax or other error in your program can lock you into a window that is to small to "poke" your way out of. As always, it is important to save programs to disk BEFORE testing them.

Load and run the program "screenform". Play with it. I think you'll like it.

Telephone 205-377-4476 Available anytime you can get me on the phone.

NOTES FROM THE UNDERGROUND

By Angel M. Matos
(c) Angel M. Matos

SE HABLA ESPANOL

FIRST THINGS FIRST

Thank you, thank you. The article in the first CBUG rag has generated a large number of phone calls. I am glad that I've been able to answer most of your questions, or at least refer you to someone who might be able to help. I also thank you for your suggestions and the information you've passed along. Again, if you'd like to call, don't hesitate, I'm happy to assist if I can; also calling me takes some of the load off of our definitely insane and slightly deranged leader, Mad Norman himself-- Norman (CBUG ? What CBUG ?) Deltzke. Now lets move on to this issue's topics.

FROM OUR LAST EPISODE:

Norman has acquired a rare beasty, a CBM 8280. A WHAT ????? It's one of the IBM-IEEE drives mentioned in the last issue. It's an IBM System 35 compatible, 8" (8" ?, that's right 8", yup !) double sided-double density (DSDD), IEEE ported, dual drive for 220 volt (European) operation. I'd love to know what the purpose of this drive, since to the best of my knowledge MS-DOS was not devised as an 8" system ?

PREVIOUS ARTICLES ON THE B/SERIES:

COMMANDER MAGAZINE//Micro System Specialties//P.O. Box 98827//Tacoma, WA 98498//206 584-6759.
**August 83, p84- "Commodore P-128" by Fred Hambrecht. --- Covers review of 'demo' P128 without final ROM; compares similar features against 8032, 4032 and C64; gives general(?) memory map. Two interesting notes appear in this article: The editor mentions that article was printed even though the P128 had shelved for replacement by the "C128" aft

intro/marketing of the B/series; and Mr. Hambrecht notes that a memory map of the 128, by Butterfield, appears in COMPUTE magazine (unfortunately, he doesn't specify issue). Mr. Hambrecht liked the P128, but found it slower than the C64 and the 4032; he reserved final judgement until he'd tested a finalized unit. Two pages.
**April 84, p70- "Introducing Commodore's B-Series Computers" by Howard Rotenberg, CBM-Canada. --- Covers general description of B/series; Use of ESC key in Basic; Use/programming of function keys; Basic 4.0 plus commands, and possible uses; ML Monitor commands. Four page article spread over six pages. NOTE: This mag died, back issues might be available ?????

COMMODORE, the microcomputer magazine//Contemporary Marketing, Inc.//1200 Wilson Drive//West Chester, PA 19380//800 345-8112.
**June/July 83, p15- "Commodore News" --- News brief describing " 'B' Series Advanced Business Computers" and noting the fact that CBM was shipping the series in all eight configurations (see CBUG#1).
**March/April 84, p18- "Commodore News -- Introducing the B-Series: Part 1" by Howard Rotenberg.
**May/June 84, p111- "user depts. -- A Closer Look at the B Series: The Machine Language Monitor" by Howard Rotenberg.
These two articles are identical to the COMMANDER 04/84 article noted above, but the layout (ergo, 'readability') is superior, four clean pages.
**May/June 84, p113- "user depts. -- Programming B128 Function Keys" by Al Fragola, Tarrant County Junior College in Hurst, Texas. Includes a short program for programming f11-f20, or f1-f20 using BLOAD AND BSAVE; allowing a maximum 1024 character for all fKey definitions, stores above 64512. Two and a half pages on three.
**July/August 84, p112- ""B Series Users Only -- Using the Commodore B128's BLOAD and BSAVE Commands" by Al Fragola. Notes for programmers on using these commands for: multi-program loading(append command), saving variables exactly as stored in RAM, windows, and program overlays. Two pages.

CURRENT COMPUTER MAGAZINES THAT MAY BE OF INTEREST.

From England--- The UK is 'Commodore Country'. CBM has a very big piece of the pie (ditto for Europe). The CBM/PET dealer network throughout the UK and Europe is very strong and has a very good reputation, so good that Apple found impossible to to challenge, and one that IBM has been surprised by--- CBM's simple PC clones are out selling Big Blue.
***Commodore Computing International(CCI)/monthly ($3.25)--- This magazine (CCI) provides a very good overview of Commodore in the UK. Covers C128, C64, and a little VIC20, C16 & Plus/4, PET. As most UK rags very heavy on the games, BUT the last six issues have shown a strong move towards productivity and programming; re: both general content and reviews. From time to time, CCI includes a free issue of Commodore Business Magazine (self contained, about 36 pages). The first two issues (July85 & Oct85) were dedicated to the Commodore PC's. The third issue (Dec85) was dedicated to the Amiga which is expected to be in distribution within the first quarter '86. It's interesting that there has been no significant coverage of the C900 (running the Coherent operating system, which is UNIX compatible) and expected to go into UK distribution in fourth quarter '85. One thing, monthly page count varies significantly, over the last six months: min. 65, max. 125.
***Personal Computer World/monthly ($2.50)--- Covers the general computing scene in the UK with a good awareness of the US and Japanese scenes. Provides a totally different and very interesting perspective of the world of computing. Commodore is covered just like any other computer company, of which there are many more in the UK than in US and Canada combined.
***Others--- The only other two Commodore oriented mags I know of are: Commodore Horizons/monthly ($2.75)--- Averaging 55 pages; and Commodore User/monthly ($2.95)--- Averaging 80 pages. Both are very heavy on the games. Programming, like business/productivity is back-of-the-book stuff, unless the programming is game related. Both are strong on VIC20 and C64, and very light on PET/CBM.

From Canada--- Again a strong PET/CBM arena, Commodore was founded in Canada (also, home of Jim Butterfield), eventually moving corporate HQ's to Wayne, PA in 1976 (West Chester, PA in 1983). Our cousins to the north provide two very interesting rags:
***The Transactor/monthly ($2.95)--- The PET lives. Very little on the B', basically a couple of notes from Liz Deal. There like here the B' was never "officially" sold, and very few machines went into circulation before the product was recalled. However, it is a nice cornucopia of all sorts of bits and pieces of info and data and stuff. This one is great for the programmer's, even though the C64 is high-lighted, other machines are covered and good solid stuff on Basic 4.0. Each issue has a central topic and averages 80 pages. Nov. 1985 was "The Languages", which includes -- A Comparison Of CPUs: The MOS 6502, Motorola 6809, and Motorola 68000 -- with a 2 page chart: Instruction Set Comparison; an article on how to convert AppleSoft Basic into Commodore Basic; and three articles covering the Intel 8088 microprocessor and MicroSoft Basic & MS-DOS as used on the CBM PC-10. Averages 80 pages. As you may of picked up I like this one, I find it very a enjoyable read. Note: this rag is issued very early in comparison to cover date, example -- I purchased the Jan.'86 issue on Nov. 20th '85.
***TPUG Magazine/monthly ($2.95)--- If you also own a SuperPet, you need this one. They've just introduced an "Super-OS/9", a very powerful implementation of Microware's OS-9 Level I (a relative of AT&T's UNIX System V). For you B' programmer's note: this system is designed for the Motorola 6809E microprocessor. Most reviews cover C64, some VIC20; the last 4 issues have been very soft on PET/CBM coverage; B' coverage: Liz Deal (strikes again)-- Spooling on the B128. Overall I have a hard time with this one; issues average less than 50 pages; issues appear on the newsstand one month after issue date, worst yet if your a subscriber, you've got to wait another 20 to 25 days; i.e. Nov. '85 issue was received Dec. 24th '85. <<TPUG has recently been suffering thru a move of their headquarters offices, assimilation of major expansions into managing several telecommunications efforts, and a re-organization of their board of directors. CBUG wishes them well and presume every probability that TPUG will soon be back on a timely course.>>

FOR THE LIBRARY

***"The Complete Commodore Inner Space Anthology" by Karl J. H. Hildon, $14.95(US) includes S/H, from: The Transactor at 500 Steeles Avenue, Milton-Ontario-Canada, L9T 9Z9 'OR' 277 Linwood Avenue, Buffalo, NY, 14209-9990. A must for CBM programmer's, "...everyTHING you wanted to know and..."; for the generalist/tech-ky it make interesting reading.
***"User's Reference Manual -- Commodore BASIC Version 4.0", First Edition -- July 1980, Commodore Part Number 321604. Not being a programmer I can't tell you if this book gives you anything new, especially if you already own the B128 Programmers Reference Guide from Protecto. But to programming dumbo me, it looks good.

PET/IEEE CABLES

Are you current cable to short ?, Do they generate to much RF interference ? Well, here are a couple of sources that may help:
***South Hill Electronics (800-245-6215, in PA. 412-341-6200) ask for "Data Communications Catalog": IEEE/IEEE (aka GPIB/HPIB) 6.6 ft. @ $44.00, 13.1 ft. @ $48.00. Also, super rare bird, IEEE Switch Boxes: 3-to-1 switch @ $109.95, 5-to-1 switch @ $149.95, X-switch @ $139.95.
***Harbor Electronics (800-243-4794, in Ct. 203-438-9625) ask for "Data Communication Interconnect Catalog": IEEE/IEEE 6 ft. @ $37.40, 10 ft. @ $40.60, fully shielded also available, call; PET/IEEE 6 ft. @ $38.90, 10 ft. @ $41.10.
***INMAC (14+ regional offices: New York-- 212-517-8015, Chicago-- 312-885-8383, Los Angeles-- 213-852-0973) ask for latest catalog: Double shielded IEEE/IEEE 6.6 ft. @ $85.00, 13.2 ft. @ $95.00.

Well, time to close out this go around. Next time we'll cover the american magazine scene and other little bites. Remember, you can contact me at: MAIL-- Mr. Angel M. Matos//3176 Decatur Ave. #6E//Bronx, New York 10467; PHONE-- Sun-Thur 8-10PM and Mon (most times) 1-5PM EST//212-231-6028; COMPUSERVE-- 75146.2224.

FOOTNOTES: 1>> Regarding the previous articles, the publishers are being contacted for Copyright releases so that we can reissue information in photocopy and/or as text files on disk formats.
2>> Two American Mags that may be of immediate assistance, COMPUTER SHOPPER/Monthly ($2.95)--- An average of 250 pages stuffed with discounted, new, discontinued, and surplus equipment. Most dealer carrying CBM/MOS chips advertise here. HOME COMPUTER MAGAZINE (503-485-9786)/10 times a year ($3.50US/$4.50CAN)--- Firstly, I don't know if this one is still alive, it is available on an irregular basis in the NY area, last issue I got was AUG85. However, if you are a programmer and can find it, programs are listed in basic and should easily convertible to B/series.
3>> Norm list me (in the JAN ScratchPad) as a Programmer --OOPS-- I'm not. I do know the basics and where to research, but realistically I am a generalist, to be more accurate, I can provide you an overview of computer systems, accessories, uses, suppliers, etc. To club members I will provide general assistance, leads, and referrals, etc. If you require professional assistance, please contact me directly; consulting services are available in the fields of: COMPUTER SYSTEMS-- for the personal and small business user; CONSUMER ELECTRONICS-- AUDIO- Stereo/HI-Fi for home and small professional installations // VIDEO- Projection TV/Monitors/VCR's for home /business // COMMUNICATIONS- ShortWave Listening/HAM Radio/CB Emergency Radio/49Mhz Talkies // TELEPHONES- Carriers: AT&T? GTE? MCI? ALLNET?/Buy? Rent?: Phones, Dialers, Answering Machines.
4>> MURA, the manufacturer of the MM-100 modem sold by Protecto and others, has a new importer as of December 1985. They are currently NOT set up to do repairs on the MM-100, they were not aware the previous importer was bringing it in. DO NOT send your modem in for repairs at this time. We are currently trying to rectify this situation ASAP, look for further info in the next SCRATCHPAD or ESCAPE, and on Compuserve/Delphi. //30//

- - - - - - - - - - SPECIAL ANNOUNCEMENT - - - - - - - - - -

Programmers' Group Forming: For those that enjoy a special challenge in programming, HELP design the "B-MASTER". This program will be designed to operate as Windowed Integrated Menu-Operated do-it-all for the the B-MEG, as well as the baby B's. The shell will automatically read the the B for memory availability, ask you how you want the memory configured- % program/% data, then it will tell you which program(s) from the Master Menu can be loaded within your memory limits, and then ask for your choice(s). The Modules to be developed will include:
*Master-Task: Will allow you to preset up to 10 Priority Tasks within a user defined period. Once set, Master-Task will give an alert to the user prior to set time, shut down (including Auto-Save) current task, load priority task and auto-executed instruction, if any. On closing priority task, you'll be returned to you original program.
*Master-Writer: Beyond SuperScript: featuring proportional spacing; double and triple column printing; page # suppression on page 1; easier printer control; more printer drivers; font driver for matrix printers; will allow importing/exporting of SuperScript and Word Result files, as well as Word Pro 8032; and more.
*Master-Base: This one will be easier and more powerful than SuperBase and will allow data import/export from/to SB.
*Master-Mailer: On special task done at its best. Compatible with Master-Base.
*Master-Calc: Will compensate for the few short comings in Calc Result.
*Master-Comm: Probably a modified version of B-Line that works within the shell.
*Master-Control: Address-Telephone Book/Auto Dialer; Memo Pad; Window to Basic; DOS Window; User defined page/window.
*Master-Accounts: Full accounting package.
*Master-??????????: User demand will dictate.

So, if you like meeting those special challenges in life, here it is. We recommend that you have a strong background in Commodore BASIC and/or Machine Language, and that you have some experience in transporting programs between Commodore machines. If you'd like to join this CBUG subgroup, drop me a note telling me what you'd like to work on, what your programming background is, etc. You can write me at: Mr. Angel M. Matos//3176 Decatur Ave #6E//Bronx, NY 10467; please mark the lower left corner of the envelope: ATTN: --CBUG--. Just as a note, I am NOT a programmer, I will be acting as the coordinator/clearing house.

PROJECT REGISTRATION REPORT

by: Marilyn Gardner
CBUG Project Registrar
1630 Madison Street
Evanston, Illinois 60202-2036

This list summarizes the registered projects either underway or completed. It is up to you to contact others working on similar projects IF YOU WISH. There is no obligation to make any contacts. The registration is for information purposes only. Your project is listed first (so I can use window envelopes). If I've described it incorrectly, please let me know. Other projects are listed in the order in which I've received the information.

Gary L. Anderson
1528 34th Street SE
Cedar Rapids, Iowa 52403 10/22/85

MEMORY EXPANSION - RAM Cartridge Circuit Board
* 24K of static RAM that plugs into cartridge slot
* two versions - fully assembled and tested or bare circuit board
* expected completion: November, 1985

MEMORY EXPANSION - One Megabyte Memory Expansion Circuit Board
* will implement Banks 0 to 14, dynamic RAM
0800-1FFF RAM or ROM
* expected completion: Spring, 1986

****

Walter Dick
Bart Bingaman
19939 Hemmingway Street
Canoga Park, California 91306 11/04/85

EXPANSION PORT BOARD - 6K RAM and Dedicated Parallel Printer Port
* expands Bank 15 $0800 to $1FFF
* uses two pins of expansion connector for memory expansion, plus one 4K X 8 static RAM
* uses one pin of expansion connector for parallel output, plus 6526A
* memory expansion completed; parallell not yet begun

MEMORY EXPANSION - 16K Add On to Calc-Result Cartridge
* 8K static RAMs work together with program ROM
* presently up and running

MEMORY EXPANSION - 24K Cartridge RAM
* Bank 15, $2000 to $7FFF
* can use Calc-Result cartridge by downloading ROM to disk
* presently being quoted for PC art and production

HEWLETT-PACKARD/COMMODORE IEEE COMPATIBILITY
* solves problems of speed differences between controllers (B is faster)
* presently running HP 2225 Think Jet on B128

CRT DIGITAL PLOTTER
* will use VIC 6500 chips to plot higher than "hi-res" graphics
* 6509 would use 16K of memory (minimum) on a B256
* will use PIA at P1 to pass graphics control functions
* presently in talking stage - problems with timing

RS232 ROUTINES FOR HAYES 300 MODEM AUTODIAL
* presently working

****

John and Kenneth Lemkelde
LemData Software
P.O. Box 175
Dover, Pennsylvania 17315 1/20/86

"BEELINE" TERMINAL PROGRAM
* Version 1.4 now available for $35.00
* handles terminal data in memory buffers and directly to disk files
* phone directory, function keys, variable format and speeds 300-9600
* extensive menus
* 15 page documentation file/manual

John and Kenneth Lemkelde
LemData Software
P.O. Box 1080
Columbia, Maryland 21044 11/18/85

"BEELINE" TERMINAL PROGRAM
* will be released in early December.

****

Fred King
King Communications
1804 Plover Spring Drive
Plover, Wisconsin 54467

MEMORY EXPANSION - One Megabyte inside B128 machine
* 960 K Banks 0 to 14
* AVAILABLE NOW for $430

DISK DRIVE CONVERSION - 8050 to 8250
* uses Tandem drives when available
* AVAILABLE NOW for $400

****

Troy Roberts
133 Pierce
Apartment 7
West Lafayette, Indiana 47906 12/16/85

PASCAL Compiler for B128
* expected completion, January, 1985
* problems now with I/O routines, floating point routines

****

Bob Anderegg
RR 2 Box 191
Loyal, Wisconsin 54446 12/24/85

DAIRY HERD RECORD Program
* health records for individuals
* breeding records
* calf due dates
* expected completion, Spring, 1986

****

Dave Leininger
P.O.Box 7345
APO New York 09012 1/11/86

DISK DIRECTORY ALPHABETIZING
* will sort directories alphabetically
* writes back to the same disk, so directory remains alphabetical
* combined with menu prograam (enhanced version of C64-Menu)
* for SFD 1001 and CBM 8050

****

Matthew Goldstein
479 Duryea Terrace
West Hempstead, New York 11552 1/21/86

COPY PROGRAM WITH HEADINGS
* allows insertion of headings to classify programs
* can change file names

MENU PROGRAM
* shows up non-selectable headings for classification

CHECKBOOK PROGRAM
* allows checks to be reconciled with bank statements

TYPEWRITER PROGRAM

CBUG NEWS - - - - FOR LOCALS ONLY

EXCERPTS FROM

Chicago B128/256 Users — December 3, 1985

March 10 - CBUG West - BASIC 4.0+ course, session 5
March 23 - CBUG East - BASIC 4.0+ course, session 5
April 14 - CBUG West - BASIC 4.0+ course, session 6
April 27 - CBUG East - BASIC 4.0+ course, session 6
May 12 - CBUG West - SUPERBASE course, session 1
May 25 - CBUG East - SUPERBASE course, session 1
June 9 - CBUG West - SUPERBASE course, session 2
June 22 - CBUG East - SUPERBASE course, session 2

CBUG West meets at 7:00 pm on the second Monday of each month with the Fox Valley Commodore Users Group at the First Congregational Church, 417 W. Main, West Dundee. From I-90, exit on Rt. 31 and go north (right) on Rt. 31 2 1/2 miles to Rt. 72. Go east (right) 1/2 mile to stoplight on 5th Street. Turn south (right) 1/4 block to entrance. Parking is in back of the church.

CBUG East meets at 2:00 pm on the fourth Sunday of each month at Bethlehem Lutheran Church, 1334 Wesley Avenue, Evanston. From I-94 or I-294, exit on Dempster Street eastbound and go east to Evanston, continuing on Dempster four blocks past the stoplight at Dodge Avenue to Wesley Avenue. Turn left (north) on Wesley one block. Church is on the southwest corner of Wesley and Greenwood. Enter on Greenwood Street. Park on street or in small lot behind church off Greenwood.

IN CASE OF BAD WEATHER: If you wish to verify that a meeting will be held, for CBUG East call Marilyn Gardner (312) 866-9159. For CBUG West (Fox Valley Commodore Users Group), call either Larry Cooper (312) 519 1514 or Herb Gross (312) 695-1316.

WHAT DO WE DO AT A CBUG MEETING?

The CBUG East and CBUG West meetings each month have the same content. It is not necessary to attend both meetings, although you may if you wish. Each meeting consists of an UPDATE on the B-series machines: new programs available, new hardware planned or available, new things which have been learned or discovered about the machines; a QUESTION-AND-ANSWER session during which anyone can ask any question about the B-series; and a FORUM which can include tutorials (see schedule of meetings for the list of coming tutorials), demonstrations, or projects. Of course, there is also the opportunity to talk with other B-series users, an opportunity which may increase the planned two-hour meeting time by 50 per cent.

DEMONSTRATIONS coming early in 1986 include DISK DRIVES in January and March - the CSI 10 megabyte hard drive, the CBM 9060, and an eight inch IBM drive. A coprocessor board, when available, which will allow the B-series to run CCPM86 and MS-DOS programs, and the one megabyte B running integrated Calc Result/Word Result.

TUTORIALS

Following the tutorials in BASIC 4.0+ and SUPERBASE listed on the meeting schedule, we will have tutorials on disk files (1 session), B's extra features (1/2 session), and Commodore printer features (1/2 session). A four-session course on 650x asssembly language programming may be offered at a time and place separate from the regular meetings, since less than a dozen people at CBUG East and West combined have shown interest in this.

Those who have been at meetings for the BASIC tutorials know that the written documentation given out at the sessions is a detailed and valuable piece of work. For those who missed the first session(s), extra copies of the tutorials will be available at later sessions. This BASIC 4.0+ tutorial will also be available on disk, offered with the first CBUG ESCAPE of 1986. Warren Swan's detailed descriptions of the specific BASIC used on the B128 are useful to all, even persons experienced in BASIC programming.

WARREN SWAN, the teacher of the BASIC 4.0 course, has been programming for 14 years. He is presently a programmer for AT&T Bell Labs. He has an M.S. in computer science from IIT, and has owned Commodore equipment since the early PETs came out in 1977. He is familiar with the B-128, having owned one since January 1985 and having disassembled the ROMs (read-only-memories, which give the computer its unique language and capabilities) soon after he brought his machine home. Lest those who have not met him think him merely a highbrow professor, Warren's credits also include an improved Star Trek game for the B-128, available on disk in the program library, and discovery of a number of interesting and fun facts about our machines.

Warren's approach to the course is a layered approach, instead of a reference book style. His aim is to give beginning programmers something they can use at once, rather than merely discussing commands in random order. The first session covered printing simple numbers and string literals, printing simple expressions, using the semicolon, statements and line numbers, the LIST command, RUN, STOP, END, CONT, GOTO, and REM.

REGISTER YOUR PROJECTS

If your are working on hardware or software and want to avoid duplication of effort, please register your projects with Marilyn Gardner by calling me at (312) 866-9159 or writing to 1630 Madison Street, Evanston, IL 60202. It is frustrating to complete work on something only to discover that another person has done the same thing. PLEASE let us know what you are doing. I will then let you know if others are working on the same project, so you can coordinate if you want to. Let's use our programming time to develop more new things, not variations of the same things.

ZIP SORT

Send in the CBUG ESCAPE coupon if you want to know who else in your area is an owner of a B-series machine.

BASIC 4.0+ COURSE OUTLINE

SESSION 2--handling data: variables; LET; variable types (real, string, integer); expressions (operators, hierarchy); cursor controls in strings; arrays; subscripts; DIM
--BASIC input: INPUT, GET
--programming: multiple statements per line; "line" length; NEW; DELETE
--program flow: IF expr GOTO line#; IF expr THEN statement (ELSE statement); FOR / NEXT

FUTURE SESSIONS--basic output: print fields and comma; TAB(x); SPC(x)
--handling data: built in functions ABS, SGN, LEFT$, MID$, RIGHT$,...
--program flow: ON expr GOTO line#, GOTO line#,...
--using disk commands: DLOAD, DSAVE (with replace), checking statuses DS$, DS, and ST
--maintaining data files: PRINT#; INPUT#; GET#; DCLOSE; CLOSE
--program flow: GOSUB / RETURN; ON expr GOSUB line#, line#,...
--handling data: READ / DATA; RESTORE; user functions DEF FN ... (X)=...;CLR
--programming: illegal direct commands (INPUT, DEF FN, etc.)
--using disk commands: COPY; SCRATCH; RENAME
--maintaining data files: CMD
--comments: after GOTO, GOSUB, THEN #, ELSE #
--basic output: PRINT USING; PRINT#..,USING
--miscellaneous: TI$; KEY; BANK; POKE; PEEK
--using disk commands: VERIFY
--TRAP / RESUME
--miscellaneous: DISPOSE; BLOAD; BSAVE; PUDEF; SYS; USR(X); WAIT

The CBUG LOCAL is prepared and distributed by Marilyn Gardiner <<Full text on print file disk, file "cbuglocal 1286">>

CONVERSION OF CBM 8032 AND C-64 BASIC SOFTWARE FOR USE ON THE B-128

by John Wolfram

Space Coast Commodore Business Machines Users Group <<SCCBMUG>>

It would seem that with so little software available for the B -128 and so much for the CBM 8032 and C-64, that a logical thought would be to convert software between the machines. This thought is not only logical, but relatively easy to accomplish, within certain limits. What are those limits ? Disk format, poke and peek codes, basic syntax, loading address, and for the very brave - conversion of machine code instructions.

This article is intended to be a beginners guide to the conversion of software. Later articles will deal with some more advanced techniques.

GETTING STARTED

In order to convert software you will need some basic tools of the trade. These include several utility programs (available from CBUG and other sources), a couple of books, and a modem or access to someone who has either a CBM 4040, CBM 2030, <<a MSD or other IEEE 1541 read compatable>> disk drive. A listing of the materials needed is contained in the bibliography at the end of this article.

The first problem is to decide upon what to convert. As a beginner, it is recommended that you begin with a program that is written in basic with few if any peeks or pokes. An obvious question, how do I find out if a program is written in basic or if it has peeks and pokes ? To do this load the program on the source machine and list it. If the program is written in basic, the program information will be displayed on the screen in orderly line numbers. Once you have determined that the program is written in basic, you should then printout a copy of it for further examination. Things to look for after listing include: SYS calls from the program, long numerical data statements which are POKE'd into memory locations, and the use of sprites (calls to C-64 memory locations 53248 - 53294). If any of these appear in the listing, select another program to convert. Conversion of programs of this type require programming skills which are beyond those of a beginner. One last item, you should also avoid programs which require the use of joy sticks, game paddles and the C-64 function keys at this early stage of your career.

THE FIRST STEP

Now that you have selected several programs for conversion, you must now deal with getting them from the source machine to the CBM 8050 disk drive supplied with your B - 128. This can be accomplished by two methods. The easiest is to use a CBM 4040 or CBM 2030 disk drive. The other is by transferring the programs between computers using a modem and terminal software.

If you can borrow a CBM 4040 or CBM 2030 disk drive (most of the 8032 and 8096 machines have these drives) the transfer process is as follows:

1. Turn off all your equipment and then connect the extra drive to the piggy back connector on the 8050 drive using a standard IEEE-488 cable.
2. Turn back on everything except the disk you just connected.
3. Load the '8432 emulator program' and run it.
4. Load the 'unit addr chng' program and run it. Change the unit address of the 8050 to '9'
5. Turn on the extra drive. It is now unit '8'
6. Load the 'scopy5' or 'copy.all.rev' <<or copy-all b128.3>> program and run it.

You are now ready to transfer your selected programs from the CBM 8032 or C-64 disks to the CBM B - 128 <<with 8050 disk drives>>. Place the disk to be copied in the 4040/2030 drive and copy it to the 8050 using the instructions for the program in use.

Since you probably have borrowed this equipment, it would be a good idea to copy several programs during this session. This will give you material to work on for several weeks and if you are unsuccessful in the conversion of some of them, the session won't be a total loss to you.

If any of the programs you intend to repair use sequential data files, be sure that you copy them to the 8050 drive along with the programs. Otherwise, when you try to run the program it will report a file not found error.

MODEM TRANSFER

If you can not find or borrow a suitable disk drive, it is possible to transfer program ano data files between systems using the telephone and modems.

In order to do this, you will need a modem and a copy of the 'X-modem terminal program' (available from several sources including CBUG). To transfer programs in this way, have a friend with the source computer call you up and then connect his modem to the line while you do the same. Using the X-modem transfer protocol, download the programs that you wish to convert. Don't forget the data files !!!!!!

<<You can connect two computers of dis-similar types together thru their RS232 ports without use of modems as well. What is known as a null modem is simply a cable with a DB25 connector at each end with pins 1 connect to 1, 2 to 3, 3 to 2 and 7 to 7. Pins 2 and 3 are the talk and listen data pins, 1 and 7 are grounds. There are rarely transmission problems in this mode as there is no phone line to pick up or cause interference.>>

Since the modem transfer sometimes creates errors in the programs, you should printout the copied program and compare it with the printout made on the source system before continuing with the conversion process. Repair any modem induced errors before continuing. It is very frustrating to spend the time to convert a program only to find that it contains errors introduced during the transfer and because of them won't run.

THE SECOND STEP

Now that you have a copy of the program on your 8050 disk drive, the biggest step has been accomplished. From here on things get fairly easy.

Before you attempt to 'fix' anything, make a backup copy of the disk or disks that contain the programs that you just copied. This way if you do make a mistake somewhere along the line, all will not be lost. You may lose some time but you will still have a copy of the program in it's original form should you need to start over from the beginning.

The road to conversion begins with the modification of the program loading address. While many of the CBM 8032 and C-64 programs will run on the B-128 without modifying the load address, it is recommended that you do modify it for system consistency and in case that you decide to operate your system with a DOS wedge or other utility present. If you don't modify the address you may have problems with programs overwriting one another. This generally leads to system crashes and a lot of hair pulling and uttering of words not suitable for the ears of small children.

To modify the loading address, the following needs to be done:

1. Load and run, Jim Butterfield's 'Disk Mod' <<or 'Chg addr 1'>>program
2. Insert the disk you are working on and enter 39,1 for the track and sector numbers. This will display the contents of the first block of the disk directory in hex code and ascii. It is suggested that you printout a copy of the screen using the proper command. If the disk contains many programs, continue to the rest of the directory blocks using the next command. Be sure to printout each block as you go.

READING THE DIRECTORY BLOCK PRINTOUTS

Looks like a bunch of Greek doesn't it ? In a few short minutes you won't think it is anymore.

If you don't know what hexadecimal numbers are and how to read and convert them, you had better go and read the reference in the bibliography about them. Come back after you know that $11,$ff really means track 17, sector 15.

The following is an example of an entry in the disk directory. It looks like this:

```
40: 00 00 82 13 06 4f 48 4d 53 20 4c 41 57 a0 a0 a0 :      ohms law
```

Let's look at each piece of this line in sequence.

1. 40: - this is the location in the directory block of this information, it is of no interest to you so forget it.
2. 00 00 - if this were the first entry in the dictionary some other numbers would be here, since it's not the 0's appear. Forget about them.
3. 82 - this is important !!!! This tells you that this is a program file and not something else. At this stage of your development, don't mess with any file that doesn't have an '82' in this position !!!!
4. 13 06 - this is the beginning track and sector number for the program 'ohms law'. It converts to track 19, sector 6 in decimal. This is the track and sector that you will need to enter in 'disk mod' to get to the location of the first block of the program to change the loading address. Write it down on the printout. Of course it will be different on you printout because you will have some other hex numbers in this position, so convert them to decimal and write down the numbers on your printout.
5. 4f 48 4d 53 20 4c 41 57 - these are the ascii hex values of 'ohms law' which is printed at the end of the line. Don't worry about them.
6. a0 a0 a0 - these are 'nulls' put there by the computer to fill up the rest of the line. Don't worry about them

Once you have found the starting address of each of the programs that you wish to modify, reload 'disk mod' and run

✱ THE CBUG LIBRARY ✱

The CBUG Library continues to grow -- at a phenominal pace. Last fall we brought you 10 disks, in January 1986 three more, and now, FIFTEEN more. But it is the quality and breadth of subjects covered that's important. No where else in the world of Commodore, are such professional level applications programs to be had on a library basis (if at all). No other organization can boast the esprit de corps that our contributors have shown -- thousands of hours of effort in writing and compiling programs; in some cases programs they've paid professional programmers huge sums to develop for their individual use!

But, you say, "I'm not a doctor or a news dealer/distributor." A physician's billing might be adapted rather easily to help with attorney's time accounting, appointment scheduling, etc. "But, a newspaper dealer's program, that's even more arcane," you plead. Not at all! There are lots of applications for route planning and analyzation -- politics, municipal planning, delivery scheduling of any kind!. One needs be far less a programmer to adapt than to write anew. A few, though, such as the series of electronics formulas are a bit specific. Either you need them or you don't. Even then, they do have value to all as studies in programming technique.

At the outset of CBUG, our objective was to bring as much information to our members in as timely a manner as possible. That includes trying to delivery nearly everything we receive thru a closing date (with certan exceptions such as materials needing organization or re-collation). We promised not to hold back for the next issue, so once again here it is, the whole thing. In many cases the disk directory titles are self explainatory. However, we simply do not have the time on an expedited basis to do descriptive directories on materials sent in. As these materials become re-collated by subject in final CBUG Library disks, then such information will be generated by our Library crew; but to delay would be to deny all the information needed now. As a result of the quantity of materials we've added a new sub-class to CBUG Releases: the CBUG M# release. The "M" prefix on the CBUG number indicates that we have relied on the contributor(s) to have provided sufficient on disk documentation for the programs and information files.

Why Pre-Release Disks? There are always materials received that are not easily classified, have limited documentation, or don't appear at first glance to belong on a higher rated Library disk, yet, some of the materials are too important to hold back from members. We've received numerous letters of appreciation from those who enjoy digging thru the unknown. Not to mention many members who have helped out by repairing and improving programs so that when incorporated in the final form, the product is further enhanced. Pre-Release disks are both fun and educational -- and sometimes invaluable.

In this issue we bring you Liz Deal's upgrade, which if you have the earlier version you can obtain the new one at 1/2 the current price by returning the special "Peel Here" CBUG label and affixing it in the space provided on the order form. The label must be returned to obtain the discount. This program applies ONLY to the "Peel Here" type labeled disks, in the case of Liz's Utilities, #12877. For accounting reasons, the half price order uses a special stock number though the copy you receive will be labeled as the main current stock number.

Members have asked about the Freeware concept, particularly as it applies to Royalty bearing disks. Sometimes authors ask for royalties which are included in the initial sale of the disk (and remitted to the author by CBUG). However, if you make a copy and give it to another member, the author has been deprived of their due royalty and it should be remitted to the auther in a sum not less than the difference between the current CBUG library price and the base $9.00 CBUG library fee. Of course, making copies for your own backup does not incur a royalty responsibility. Freeware is a similar concept based on your "liking" or "putting to use" a program(s) rather than just acquiring a copy of it. Some authors state what they feel is a minimum sum to be paid, others leave it up to the beholder. Whatever the case, or if you merely wish to thank one of our authors, let them know what you think of their programs, give them suggestions, and if appropriate say thanks with a few bucks.

The operations of CBUG and the Library are in substantial part supported by some of the library revenue. Though most CBUG library materials are permissibly copied amongst members, CBUG is deprived of some operating revenue by each non-CBUG made copy. Therefore I've requested members send in $5.00 for each copy of CBUG library disks made for another member.

Occasionally members will damage library disks before making a backup. CBUG will be happy to re-duplicate or replace the disk as necessary. Kindly enclose $5.00 to cover our handling, postage, etc. Return either the disk or the label from it with an explainatory note. We'll try to do it within a couple of days. This applies ONLY to library disks, NOT to commercial products CBUG may distrubute such as Physical Exam etc.

Always make backups before doing anything else with a disk. None of the CBUG Library disks are copy protected. However, on occasion our duplication drives may be marginally out of speed. Under some circumstances, if your drive should be off speed in the opposite direction you may be unable to make a back up. If so, try making a backup in the other direction (i.e. d1=0 or d0=1) if that fails, verify that you are able to back up at all with something you know did work. If it is not your system, send us a note and we will send you anther copy gratis.

With this publication we have entered yet another area of copyright distinctions. A substantial number of materials offered are provided for **the sole use of CBUG members** and all other disemination is strictly prohibited. These files are marked in their contents or labeling as such. The CBUG Archive listings are a most notable example of that situation -- repleat with extensive disclosure agreements. There are several articles in this publication that appear on disk as well -- or full text on disk, that are for CBUG membership use ONLY. Often such copyrighted or restricted offerings are offered elsewhere on a commercial basis. It is imperative that all members follow the letter of these notices lest we loose such exceptional opportunities.

For those of you still having some difficulty knowning how to load files: the disk directory is obtainable from basic by hitting F8 RETURN; in Superscript also by hitting F8 followed by Drive # and RETURN (exp: F8 0 RET). All files list their type to the right of each directory line. Generally speaking SEQUENTIAL files can only be loaded from properly equiped programs such as Superscriptp, terminal programs, etc. PROGRAM files on the other hand are generally intended to be loaded ONLY in basic. A notable exception is program files that are subsidiary to others, as in a multiple program program -- one with an opening basic program (which you load and run), followed by several machine language programs which are loaded by the machine from the first basic "loader" program. Another exception is program files run in Superbase which are stored as sequential files with a ".p" suffix on each file name. Loading of these will produce garbage, nothing, or lock up your computer. Program files may be loaded by simply hitting F3 followed by the "filename" and RETURN. Then when the "ready" re-appears, type RUN. The basics of this are covered in the two thin book which came with your computer. You may also refer to some primer books at the library or book store on the VIC20 or C-64 looking for chapters dealing with disk instructions. They are all about the same, but ours are much easier as the B128 has a built in "dos wedge"

A NOTE TO FUTURE CONTRIBUTORS: CBUG would appreciate receiving an anotated directory file not to exceed 73 columns in width as the last file on each disk. These files should be in Superscript, and take the following form as example:

```
10  "one.p"          seq  Sets up/extracts account data for invoice
19  "cooper RS232"   seq  Basic discussion of RS232 modem connection
```

The next to the last file on the disk should be a brief "blurb" explaining the files contributed -- your suggestion of what we should list above the directory in the library listings.

Those of you who have already contributed disks without this type of information, are requested to submit just this information on a separate disk as soon as possible. Conspicously mark this new disk as "ANOTATED DIRECTORY FOR: xxxx ", rather than a new submission. We'd like to put out a complete fully explainatory catalogue maybe by late summer if we can get all the information this spring.

I know I've been rude in not acknowledging receipt of disks as they come in. With the press of CBUG business, submissions go thru a cumbersome proceedure which does call for sending providers a package of blanks -- but this process occurs during the heat of the publishing cycle, so the actual sending is shortly after the publication date. It's not that we're ungrateful, just a mite busy. Please accept my appologies.

LIKE WOW -- Look what we've got for you Superbase users and would be users. THREE new disks, the beginnings of the CBUG Superbase Suite.

NORTHROP'S SUPERBASE AIDS CBUG #13 **NEW RELEASE** #12787
LEIGHFIELD ON SUPERBASE

Did you know that you can hook up 4 8050 drives to your B128, each with a different Database in it, and write a program in Superbase that will allow the accessing of not only 4 drives, but manipulation of data between 4 different databases? This is just one of the startling revelations Mr. Northrop discloses in his Superbase aids sample programs and numerous help files. Many of the interesting nuances of Superbase are disclosed and made easy.

Brian Leighfield is a staff programmer with Precision Software of London. the publishers of Superbase and Superscript. Brian has provided us with a series of disertations of superbase helps and aids, totalling 60 pages. These are absolutely invaluable morsels of information which just on their surface answer about 90% of the despiration calls I've received regarding Superbase.

One of our local members upon reviewing Mr. Leighfield's text writes: Here is the definitive tutorial on hard-to-do things in Superbase. You must be familiar with programming in SB to utilize all the information in the disk, but if you have ever agonized over label making with SB; the answers are here; if you think that the troubles you have had transferring files from one disk to the other may be related to hardware; the way to find out and the remedies are here. This is real good stuff if you are a SB user.

The materials hereon are copyrighted by Messrs Northrup and Leighfield as to their own contributions on this disk. Permission for library distribution within and amoung CBUG members is the only lisence given.

```
1 "Superbase Helps " sh 2c  8  "create & post.p"   seq  1  "hHow to Use"        seq  10  "accounts"          seq  20  "fields&calcs"       seq
2    "hSuperbase Helps" seq 4  "create calendar."  seq  5  "end-month.p"        seq  1   "CLUB"              seq  60  "command tricks"     seq
5    "hDual Database"   seq 5  "autoCreateMonth."  seq  4  "mid-month.p"        seq  2   "hAbout CLUB"       seq  8   "filedefcorrupt"     seq
2    "hDrive 1 save"    seq 5  "post calendar.p"   seq  6  "statements.p"       seq  3   "menu.p"            seq  47  "duplicate keys"     seq
5    "hCommands"        seq 6  "start.p"           seq  4  "check report.p"     seq  2   "hRead Me First"    seq  17  "keys"               seq
4    "hWhy!!!"          seq 3  "dir.p"             seq  7  "month report.p"     seq  1   "END NORTHROP"      seq  98  "drives/filecopys"   seq
3    "hDirectory Help"  seq 3  "hCalendar"         seq  2  "enter address.p"    seq  3   "---above---"       seq  41  "superofficing"      seq
6    "hStart Gold"      seq 3  "hcalendar"         seq  6  "report by name.p"   seq  4   "-- CBUG MEMO --"   seq  13  "S&R Instructions"   seq
3    "h---FUN STUFF---" seq 1  "---Tip or Two---"  seq  2  "select label.p"     seq  101 "labels"            seq  1344 blocks free.
1    "CALENDAR"         seq 3  "hCreate Director"  seq  5  "checking"           seq  84  "reports"           seq
6    "month"            seq 2  "hBack to Dir"      seq  1  "---SAMPLE---"       seq  63  "progs&menus"       seq
```

MEDICAL ACCOUNTING w/ SUPERBASE CBUG #14 NEW RELEASE #12684

Superbase Physician Billing for HCFA-1500 Insurance Forms. "Set.up" is a collection of 5 programs written to run within the superbase environment. It allows any of four variations of the HCFA-1500 form to be printed by pressing a key f1-f4. The system configuration as called for currently is the basic Protecto set, using the 8050 disk drive. A version using the CSI ST10C 10 MB disk drive is in progress now.

The programs were written for Dr. Adrian M. Griffin,MD. This gentleman is certified as both an Emergency Medicine Physician and a Psychiatrist. He has an in-hospital practice within the Emergency Department of our local hospital, with associated follow-up with his private office. Additionally, he has his Psychiatry practice through his private office, as well as staff priviliges to admit and care for his patients as the staff psychiatrist. This multiple skill conjours up a dragon when it comes to the standard billing protocol with good ol' HCFA-1500. He has two different provider numbers, plus either of the two must be prefixed with an "89" for medicaid patients. Hence: "Set.up" , which allows his secratary to use function keys to print a single or multiple HCFA forms for either his ER or his PSY practice.

The single form filer allows just that, for either a solitary patient treatment, or to refile one of those claims rejected without explanation by one of the MEDIs. Or, after entering 15-20 patients into the hcfa file, seen the night before in the emergency room, the batch function allows the secratary to go for coffee while the B128 prints out all 20 insurance forms on the printer.

The PSY programs are most suitable for use with a private practice, and can be easily modified with the SUPERBASE program functions to contain the user specific information such as name, provider number, address, etc. The ER programs are suitable for a hospital practice with separate billing.

The database name is "pt files" and includes data files called "hcfa", "pt info","pt dx", and "pt acct". The prime file, of course is "hcfa". Non-insurance data of a privileged nature is stored within "pt hx" and "pt meds" for those purposes.

Mr. Robert Lackey is the author of these programs and advises that he is prepared to assist practicioners in modifying the program to use variants of the HCFA forms. He has written clerical functions for radiology departments and offies, patient scheduling for private offices, etc and would consider converting these to the B128 for a nominal fee should interest be shown.

This disk requires payment of $15.00 royalty if put to commercial use by a CBUG member. The programs and text hereon are copyrighted by Mr. Lackey, and all reproduction or use outside CBUG membership is prohibited.

```
1 "AGdBase         " 1a 2c  4  "hbatch"       seq  1   "TRAINING"         seq  1  "setup.p"      seq  10  "singleerhcfa.p"    seq
6    "start.p"          seq 5  "hsort"        seq  1   "PT"files          seq  5  "singleer.p"   seq  10  "singlepsyhcfa.p"   seq
4    "henter"           seq 4  "hprog"        seq  10  "pt acct"          seq  6  "hcfaer.p"     seq  11  "batchhcfapsy.p"    seq
5    "hselect"          seq 4  "hmaintain"    seq  4   "pt info"          seq  1  "pssetup.p"    seq  11  "batchhcfaer.p"     seq
8    "hfind"            seq 3  "hmemo"        seq  1   "hlist"            seq  5  "singleps.p"   seq  0   "backup"           *prg
5    "houtput"          seq 7  "hmenu"        seq  1   "olist"            seq  6  "hcfaps.p"     seq  10  "hosingle.p"        seq
5    "hcalc"            seq 3  "hcommands"    seq  1   "klist"            seq  8  "hcfa1"        seq  11  "hobatch.p"         seq
5    "hreport"          seq 3  "himport"      seq  2   "pt dx"            seq  1  "CONSULT"      seq  10  "posingle.p"        seq
4    "hexecute"         seq 3  "hexport"      seq  1   ""contreat"        seq  8  "hcfa"         seq  11  "pobatch.p"         seq
5    "hhelp"            seq 7  "hlabels"      seq  1   "Contreat"         seq  1  ""hcfa"        seq  3   "-----rl-----"      seq
5    "hfile"            seq 2  "cust.rec"     seq  6   "batchform1.p"     seq  9  "erhcfa.p"     seq  30  "set up instr."     seq
7    "hformat"          seq 2  "cust.inv"     seq  6   "batchform1bu.p"   seq  1  "set.up.p"     seq  1714 blocks free.
```

FRIENDFAM CBUG #15 NEW RELEASE #12716

FRIENDFAM 2000 is a full-featured information system geared to keep tabs of all of your friends and family in a very organized, useful, and fun manner. IT DOES REQUIRE SUPERBASE. After 'booting' the B-128 with Superbase, replace the Superbase program disk with the FRIENDFAM disk and press RETURN. You will then be presented with the FRIENDFAM main menu, and you will be ready to go!

Entering data into FRIENDFAM is a breeze. In fact, when I have 'unregistered' friends or family visit my house, I have them create their own FRIENDFAM record on the system, and I then show them how I use the information for very useful purposes, namely printing or displaying the following: 1) up-coming birthdays; 2) up-coming anniversaries; 3) Xmas cards sent; 4) Xmas cards received; 5) Address list; 6) phone list; 7) FRIENDFAM roster. It's great fun, and everybody likes being part of the data base!

For you Superbase programmers (and would be Superbase programmers!), FRIENDFAM 2000 offers many good examples of structured code which can be used for reference. For you non-programmers, FRIENDFAM offers a meaningful application for use with Superbase. Mickey Crittenden.

FRIENDFAM is copyrighted by Mickey Crittenden. Reproduction beyond CBUG membership is prohibited without prior permission.

```
1 "workdisk1       " aa 2c  2  "notes.p"           seq  1   "TRAINING"     seq  1  "pdatarpt.p"     seq  2   "pmailist.p"        seq
19   "start.p"          seq 2  "pnotes.p"          seq  1   "FRIENDFAM"    seq  2  "pbdayrpt.p"     seq  2   "phonelist.p"       seq
3    "annivers.p"       seq 1  "FROM"              seq  2   "persondata"   seq  3  "tsort.p"        seq  2   "pphonelist.p"      seq
5    "hselect"          seq 1  "1 fill 1"          seq  2   "hlist"        seq  2  "cardsend.p"     seq  7   "labels.p"          seq
2    "panniver.p"       seq 1  "additional info"   seq  1   "datarpt..p"   seq  2  "pcardsend.p"    seq  12  "superbase primer   seq
2    "pannivers.p"      seq 6  "friendfam.synops"  seq  2   "datarpt.p"    seq  2  "cardrecvd.p"    seq  1878 blocks free.
4    "hinstruct"        seq 11 "sb.primer"         seq  19  "tstart.p"     seq  2  "pcardrecvd.p"   seq
5    "hstart"           seq 1  "1 fill 5"          seq  2   "bdayrpt.p"    seq  2  "mailist.p"      seq
```

SUPERBASE APPLICATIONS PROGRAMS #1 CBUG #7 #12932

Rev Clyde Northrop must surely be getting some extra help. We have here a complete business suite to run a furniture store -- which he wrote for his brother. Dan VanWeirengen shows us just how easy it is to make the B128 handle data bases of inordinate size. Last but not least is Jim White's important revised labels program for Superbase -- which allows for putting multiple fields on one line (i.e. CityState), and putting punctuation & spaces between them (City, State). Doyle Horine has provided a General Ledger program which circumvents many of the shortcomings of Superbase as to lengthy calculation strings, etc. The beginnings of a nice collection.

NOTE: Clyde's is offered as Freeware for casual use. Permission is required for commercial use.

```
1 "CBUG SUPERBASE 1" 93 2c  1  "fake"          seq  11  "five.p"           seq  11  "monthly post.p"     seq  19  "ss.directory"       seq
14   "ss.datab.sampler" seq 1  "FURNITURE"     seq  18  "daily ledger.p"   seq  1   "stock printout.p"   seq  2   "--id furniture--"   seq
1    "date"             seq 1  "fake 2"        seq  13  "enter stock.p"    seq  1   "CHECKING"           seq  7   "---labels---"       seq
```

```
15  "ss.ref.superbase" seq  1   "fake 3"          seq  2  "inven hlist.p"    seq  5  "credit u"          seq  8   "labels.p"           seq
8   "start.p"          seq  1   "create.p"        seq  2  "inven alist.p"    seq  5  "bank"              seq  1   "---hospital---"     seq
11  "accounts"         seq  10  "one.p"           seq  2  "inven by brand.p" seq  2  "hlist.p"           seq  22  "september85"        seq
7   "invoice"          seq  15  "two.p"           seq  4  "inventory report" seq  8  "credit u.date.p"   seq  1   "-general ledger-"   seq
8   "ledger"           seq  15  "three.p"         seq  4  "menu 3.p"         seq  8  "report by #.p"     seq  20  "setup.p"            seq
5   "inventory"        seq  6   "four.p"          seq  3  "menu 4.p"         seq  8  "report by date.p"  seq  17  "print.p"            seq
                                                                                                            1575 blocks free.
```

```
brief: A Directory of Superbase Sample Programs by Clyde Northrop
   This disk created for my brother.
 ALL FILES PRECEEDED BY ss. SHOULD BE LOADED WITH SUPERSCRIPT II         18 "daily ledger.p"   seq  Keeps a daily ledger of department sales.
14 "ss.datab.sampler" seq  Brief instructions. Read with Superscript II.  13 "enter stock.p"    seq  Fast entry program to post incoming stock.
1  "date"             seq  Contains the data of Last Start, etc.          2  "inven hlist.p"    seq  Sample holding list for inventory report.
15 "ss.ref.superbase" seq  Info in Superbase. Read with Superscript II.   2  "inven alist.p"    seq  Sample holding list: Alphabetical inventory.
8  "start.p"          seq  A modified Superbase start program.            2  "inven by brand.p" seq  Sample holding list: Inventory by brand.
11 "accounts"         seq  Format for customer accounts.                  4  "inventory report" seq  Report Program for inventory.
7  "invoice"          seq  Format for customer invoice.                   4  "menu 3.p"         seq  Menu for special functions.
8  "ledger"           seq  Format for daily ledger.                       3  "menu 4.p"         seq  Menu for sepcial functions.
5  "inventory"        seq  Format for inventory.                          11 "monthly post.p"   seq  A monthly report program.
1  "fake"             seq  A dummy file.                                  1  "stock printout.p" seq  Simple stock list program.
1  "FURNITURE"        seq  The database.                                  2  "hlist.p"          seq  A way to look at an output list.
1  "fake 2"           seq  A dummy file.                                  The following Database is Loaded from Superbase Menu 1 or 2 by typing:
1  "fake 3"           seq  A dummy file.                                  1  "CHECKING"         seq 'database'<RET> 'checking'<RET>
1  "create.p"         seq  Creates the next invoice with no. & date.      5  "bank"             seq Format for a bank checking account.
10 "one.p"            seq  Sets up/extracts account data for invoice.     5  "credit u.         seq Format for credit union checking.
15 "two.p"            seq  Writes invoice and lists item & price.         8  "credit u.date.p"  seq Program to balance credit union account.
15 "three.p"          seq  Checks for accuracy and prints invoice.        8  "report by #.p"    seq Program to balance bank acct by check sequence.
5  "four.p"           seq  Updates stockage for perpetual inventory.      8  "report by date.p" seq Program to balance bank acct by date sequence.
11 "five.p"           seq  Posts accounts for charge and cash sales.
```

(More room can be obtained by placing "FURNITURE" database on a disk by itself and placing programs in drive 1 while database functions in drive 0.)

SWAN'S BASIC　　CBUG #16　　NEW RELEASE　　#12773

This disk contains a complete tutorial of Commodore BASIC 4.0+ for the B Series machines. It is NOT some incomplete translation of a tutorial for some other machine (like some of the programs we've all bought). This tutorial was written for YOUR wonderful little B machine.

This 6 chapter, 82 page (8.5" x 11") tutorial does NOT just teach BASIC, it teaches OUR BASIC. It teaches you what you can do with YOUR machine. It DOES teach BASIC using a "layered" approach. Concepts are layered one on top of the other - rather than presenting BASIC in reference book style. You'll be able to do some useful things with your machine almost immediately. More complex and more obscure topics are presented later, not while you're just getting used to BASIC. This building of complexity from the fundamentals inspires confidence in you, the reader, rather than leaving you confused. Plus, the tutorial contains information that even Commodore doesn't know about.

The disk contains the outline and 6 chapters in 2 forms: (1) ready to print versions - for those of you without Superscript II (tm of Precision Software), or who just want the standard print format; and (2) Superscript II files for those of you who want to print the tutorial in their own format.

The author of this tutorial is eminently qualified. He has owned Commodore machines since their inception in 1977; has been programming for 14 years, 12 of them in BASIC; and has taught Computer Science part time at Illinois Institute of Technology where he received his MS in CS. Owners of older Commodore machines may remember seeing some of his articles in other publications. He is currently the Question & Answer Coordinator for CBUG.

These materials are copyrighted by Warren Swan. Distribution beyond the CBUG membership is expressly prohibited without prior permission.

```
1 "swan's basic     " 77 2c   132  "& basic tut2"   seq   157  "& basic tut5"   seq   92   "&&basic tut1"   seq   164  "&&basic tut5"   seq
16  " start up":          prg  101  "& basic tut2a"  seq   125  "& basic tut6"   seq   246  "&&basic tut2"   seq   130  "&&basic tut6"   seq
27  "&+basic tut+"        seq  132  "& basic tut3"   seq   3    "/transfer"      prg   140  "&&basic tut3"   seq   8    "& basic tut ad" seq
86  "& basic tut1"        seq  161  "& basic tut4"   seq   27   "&&basic tut"    seq   168  "&&basic tut4"   seq   137 blocks free.
```

SWAN'S UTILITY　　CBUG #3　　#12881

Here we have a collection of exceptionally valuable disk utilities. "!futility" is a whole series of things written just for the B128, many never seen before for any machine. Such things as a program that will RESCUE a file that has been accidentally scratched -- or completely scratch a program. Recover directories and other such goodies.

Mr. Swan has provided a number of fun things too. His Star Treks, w & w/o klingons is said to be many times over more advanced than others for any computer. It uses the multiple windowing techniques unique to the B128. Not only is it a fine game, but a study in advanced programming. Have you ever wanted to have your computer "talk" back to you -- well from what I gather from the documentation, Eliza is just that.

The rest of the programs should be self explainatory from their titles -- and we've even been provided yet another top flight terminal program -- one that Mr. Swan uses in the course of his day to day work for Bell Labs as a system analyst.

Dale Finkey's backup programs are totally new and unique. Written first and only for the B128. With our high density disks (502 Kbyte), it becomes easy to archive huge quantities of data. BUT, when you copy whole disks of files to another and accumulate them, what happens if an additional disk has a file of the same name as already on the destination disk? DISASTER, and a whole big mess to clean up. Dale's program allerts you to the problem and gives you choices before the mess occurs; also you can prefix all files with the disk identifier (the two digits after the comma on the disk header) of the source disk being copied automatically. Real handy. Persuit allows the location of the tracks and sectors on which specific information is stored -- for advanced disk manipulation.

Folks, here we have a real example of some heavy duty programming written just for the B128. Soon everyone will be having to convert FROM the B128 to get the hottest stuff!

```
1 "SWANs UTILITY #1" 88 2c   31  "&&calculator"       seq  47  " star trek 1":      prg  1   "<- utilities -->"  prg  8   " bouncing ball":     prg
29   " start up":        prg  19  "& regression"       seq  4   " star trek 2"       rel  45  "!futility":        prg  1   "<communication->"    prg
1    "<-- articles -->"  prg  22  "&&regression"       seq  4   " star trek 1"       rel  1   "/count"            prg  26  " terminal":          prg
34   "& file prefixes"   seq  26  "& archive 1"        seq  8   " eliza":            prg  34  "!copy":            prg  5   "/term"               prg
36   "&&file prefixes"   seq  28  "&&archive 1"        seq  17  "&eliza data"        seq  3   "/transfer"         prg  11  " obj-to-prg":        prg
18   "& cursor config."  seq  26  "& get filenames"    seq  17  " eliza data prgm"   prg  5   " rename disk":     prg  7   " prg-to-obj":        prg
18   "&&cursor config."  seq  31  "&&get filenames"    seq  23  " iq test":          prg  34  " archive 1":       prg  2   "-----df-----"        seq
27   "& program lines"   seq  4   "& get mach lang"    seq  1   "&iqtest data"       seq  1   "/count & copy"     prg  59  "backup2"             prg
27   "&&program lines"   seq  5   "&&get mach lang"    seq  8   " craps":            prg  12  ".get filenames"    prg  7   "file pursuit"        prg
1    "<-instructions->"  prg  17  "& large letter"     seq  1   "<- demos/math ->"   prg  1   ".get mach lang"    prg  22  "backup1"             prg
97   "& star trek"       seq  19  "&&large letter"     seq  29  " lifetime":         prg  1   "<-- graphics -->"  prg  1   "-----docs-----"      prg
106  "&&star trek"       seq  58  "& terminal"         seq  10  " calculator":       prg  20  " large letter":    prg  8   "backup1.doc"         seq
20   "& eliza"           seq  62  "&&terminal"         seq  2   " factors":          prg  14  " calendar":        prg  9   "backup2.doc"         seq
22   "&&eliza"           seq  1   "<--- games ---->"   prg  13  " regression":       prg  8   " s-enterprise":    prg  4   "file pursuit.doc"    seq
29   "& calculator"      seq  52  " star trek 2":      prg  3   " squirm":           prg  4   " p-enterprise":    prg  8   "why backup"          seq
```

LIZ'S UTILITY v1.2　　CBUG #17　　UPGRADED RELEASE　　#12670

This is the first of our upgrade release program. Those of you who purchased Liz's Utility V1.1 can upgrade that disk at half the current price to this version by including the Peel Here label from the disk with this order and using part #12999. The upgraded disk in turn may be upgraded later if future versions are offered.

Liz's utility collection is amongst the most valua[illegible] you can have. From disk copy-all programs to editors, none better are to be found anywhere.

Liz has explained upgrades in full starting on page 10 of the articles in this publication. Some materials on this disk copyrighted by Liz Deal; permission required to reproduce outside CBUG.

```
1 "liz util1.2 2/86" 86 2c  46  "supermonb v7.ins" prg   17  "reader/writer.3"  prg   3   "waitline"         prg   81  "8432.21"          prg
34  "w.contents"      seq   1   "<---- b ---->"    prg   1   "<---- d ---->"    prg   8   "words match"      prg   4   "bank F.20 '1024"  prg
6   "pre-superscript" prg   4   "keytrix v7"       prg   24  "b128<-->amiga.2"  prg   27  "w.exp15"          seq   5   "instructions8432" prg
15  "b128 irq rate"   prg   16  "+b1 kxb128.v7"    prg   2   "+b128<>am.2"      prg   1   "<---- e ---->"    prg   9   "supermon4.rel"    prg
1   "+ta irq.400"     prg   2   "+b15kxb128.v7"    prg   1   "<---- g ---->"    prg   22  "list all/jim b"   prg   5   "aid4.6b00"        prg
7   "spool b128"      prg   2   "+b15kxb128.v7+"   prg   4   "picdesign"        prg   4   "cross.b128-1.1"   prg   5   "aid4.7b00"        prg
10  "w.b128 spool"    seq   2   "+b15kxb128.v7++"  prg   7   "4023 scr dump"    prg   5   "+cross.b128"      prg   6   "cross"            prg
1   "<---- a ---->"   prg   16  "+b1 kxb256.v7"    prg   8   "p.dump ins"       prg   12  "twin bagels"      prg   1   "<---- i ---->"    prg
21  "dcl b128.pal"    prg   2   "+b15kxb256.v7"    prg   5   "p.cats"           prg   23  "grungy towers"    prg   6   "peeker 80 col"    prg
2   "+btxfer.fe48"    prg   2   "+b15kxb256.v7+"   prg   1   "<---- f ---->"    prg   2   "impossible"       prg   5   "errors 8050"      prg
1   "+xcall.fe00"     prg   2   "+b15kxb256.v7++"  prg   1   "configure b128"   prg   16  "stepwise"         prg   10  "backup 8050"      prg
6   "xcall"           prg   48  "keytrix v7.ins"   prg   2   "mykeys"           prg   3   "step illustr"     prg   5   "seek id's 8050"   prg
80  "w.b128 xcalls"   seq   1   "<---- c ---->"    prg   2   "datamaker"        prg   6   "smoothing"        prg   9   "vir dir 8050"     prg
1   "<---- h ---->"   prg   4   "change dev#"      prg   4   "verifizerb128"    prg   15  "diary"            prg   8   "confirm 8050"     prg
4   "supermonb v7"    prg   23  "copy-all b128.4"  prg   5   "stringthing b128" prg   2   "events"           seq   8   "8050 errors/2"    prg
16  "+smb v7.0400"    prg   2   "+copy-a.21c3c"    prg   4   "qplot.b128"       prg   1   "<--8432 utl->"    prg   21  "clone 22 & 23"    prg
16  "+smb v7.e000"    prg   8   "disk view.b128"   prg   3   "primes"           prg   3   "Start8432.21"     prg   1   "<---- j ---->"    prg
16  "+smb v7.0400+"   prg   11  "disk logger.b128" prg   11  "squares"          prg   2   "t1"               prg   125 "w.bug2/86"        seq
16  "+smb v7.e000+"   prg   1   "+disklog.400"     prg   16  "rotate"           prg   2   "td.alt"           prg   1   "<---- k ---->"    prg
16  "+smb v7.0400++"  prg   7   "disk mod.b128"    prg   5   "tab"              prg   2   "t2"               prg   960 blocks free.
16  "+smb v7.e000++"  prg   4   "big rel file"     prg   4   "sm sort"          prg   2   "td"               prg
```

CBUG GAMES & EDUCATION #1

CBUG #18 NEW RELEASE #12792

Now we know what our librarian has been doing of late. Games, games and more games. He's figured out the B128 can be fun too! Of course he's been back down to business for a number of weeks since he completed this compendium, so its our turn now. Best yet, the games all work, and all but four return to the SHIFT/RUN menu. Mark has made it so easy even a small child can use the B128! Now this is one case where an annotated directory would generally be superfluous. Such a deal!!

```
1 "games & ed 1     " 79 2c  21  "super lander-v"   prg   3   "reaction time"     prg   1   "counter"            seq   3   "sb- blast off"    prg
1    " Startup"          prg  61  " king"            prg   106 "create a story"    prg   26  " dungeon"           prg   8   "dragon slayer"    prg
3    "  directory  "     seq  23  "solitaire-c"      prg   6   "practical joke"    prg   10  "europe quiz-e"      prg   8   "adventures"       prg
6    " menu"             prg  20  " black friday"    prg   7   "artillery-a"       prg   25  "backgammon"         prg   8   " advscrn"         prg
13   "chomp"             prg  17  "concentration-b"  prg   8   " instscrn"         prg   30  " explore"           prg   116 " wisp"            prg
23   "afo/balloon-v"     prg  12  "deflection-v"     prg   17  "sb- organ"         prg   19  "backgammon instr"   prg   1   " advf 1"          prg
27   "!!--NOTICE--!!!"   seq  55  " nuke"            prg   71  " oregon trail"     prg   28  "backgammon tutor"   prg   8   " bfscrn"          prg
1    "average"           seq  59  " atlantis"        prg   16  "president's quiz"  prg   68  "instructions"       seq   8   " nscrn"           prg
33   "hangman"           prg  12  "yahtzee b128"     prg   11  "percent drill-e"   prg   19  "hangmath- e"        prg   8   " ascrn"           prg
20   "blackjack-c"       prg  10  "lottery"          prg   18  "q's & z's -e"      prg   8   "hi-lo"              prg   8   " otscrn"          prg
14   "bomber-v"          prg  5   "lotto"            prg   16  "reverse- e"        prg   8   " wwscrn"            prg   8   " dscrn"           prg
27   "checkers"          prg  3   "kalidascope"      prg   8   " bootscrn"         prg   2   "sa-sound & the b"   prg   8   " escrn"           prg
19   "dominoes"          prg  17  "craps game"       prg   19  "big math-e"        prg   4   "sb- tune gen."      prg   1   "bets"             seq
15   " hamurabi"         prg  25  "therapy-a"        prg   26  "blackbox-v"        prg   3   "sb- halley's c"     prg   8   " ktscrn"          prg
27   "yellow light-v"    prg  28  "dog bite"         prg   7   "decimal / -e"      prg   6   "sb- yankee do"      prg   1   "winnings"         seq
54   "labyrinth-va"      prg  14  "caves of ice-av"  prg   5   "density-e"         prg   8   "sb- amaz grace"     prg   144 blocks free.
27   "lunar lander-v"    prg  5   "numbers game"     prg   12  "division drill-e"  prg   7   "sb- harmonica"      prg
```

MARLIN & FRIENDS WRITINGS

CBUG #19 NEW RELEASE #12805

And again, was our librarian hard at work. This disk the the entire text files of all publications of the deceased B128 Users Group of Lompoc California. Also included are those programs that were on the utility and game disks offered by that organzation which CBUG has not yet published elsewhere. For those of you who did not receive your issues, which ran from December 1984 thru May 1985, here they all are. For others, this is yet another wealth of assorted information. CBUG is in no way related to, nor is CBUG successor to Marlin's effort; this disk is brought to you as a courtesy and in the spirit of educational disemination of information on which CBUG is based.

```
1 "B128 Journal    " j1 2c  2   "bug1/3/014"   seq   13  "bug1/5/004.1"  seq   2   "bug1/1/006"          seq   66  "kernal-b128.1"     seq
3    " loader"          prg  16  "bug1/4/cov"   seq   12  "bug1/5/005.1"  seq   7   "bug1/1/007"          seq   55  "kernal-b128.4"     seq
6    " menu"            prg  12  "bug1/4/002"   seq   9   "bug1/5/008"    seq   12  "bug1/1/008"          seq   51  "kernal-b128.2"     seq
7    "diskmod"          prg  9   "bug1/4/015"   seq   9   "bug1/5/009"    seq   10  "bug1/1/009"          seq   59  "kernal-b128.5"     seq
6    "northern tune"    prg  10  "bug1/4/006"   seq   9   "bug1/5/010"    seq   11  "bug1/1/010"          seq   1   "ctl.kernal"        seq
17   "b-128 organ"      prg  10  "bug1/4/007"   seq   8   "bug1/5/011.1"  seq   7   "bug1/1/011"          seq   39  "asm"01/17/83       prg
12   "dirgen.p"         seq  12  "bug1/4/004"   seq   8   "bug1/6/004"    seq   9   "bug1/1/012"          seq   17  "b128.defs"         seq
14   "screen editor"    prg  11  "bug1/4/005"   seq   10  "bug1/6/005"    seq   14  "bug1/1/013"          seq   5   "b128.linput"       seq
2    "  directory  "    seq  9   "bug1/4/008"   seq   10  "bug1/6/006"    seq   13  "bug1/1/014"          seq   3   "terminal"          prg
8    " instscrn"        prg  9   "bug1/4/009"   seq   9   "bug1/6/007"    seq   10  "bug1/1/015"          seq   8   ").$%8"             seq
9    " bootscrn"        prg  9   "bug1/4/010"   seq   11  "bug1/6/003"    seq   12  "bug1/1/003"          seq   21  "fkey editor"       prg
27   "!!--NOTICE--!!!"  seq  11  "bug1/4/011"   seq   24  "bug1/6/007.1"  seq   8   " cbugboot"           prg   36  "address 128"       prg
26   "bug1/5/006"       seq  13  "bug1/4/003"   seq   11  "bug1/6/008"    seq   8   " cbuginst"           prg   1   "ascii code no."    prg
10   "bug1/2/cov"       seq  10  "bug1/4/014"   seq   10  "bug1/6/009"    seq   3   "80col maze"          prg   2   "bank poke check"   prg
12   "bug1/2/002"       seq  14  "bug1/5/cov"   seq   2   "bug1/6/010"    seq   4   "change dev#.1"       prg   1   "binary-decimal"    prg
10   "bug1/3/006"       seq  11  "bug1/5/002"   seq   15  "bug1/6/002.1"  seq   13  "datamover"           prg   32  "calendar b128"     prg
11   "bug1/3/007"       seq  14  "bug1/5/003"   seq   9   "bug1/6/016"    seq   11  "pet datamv.pal"      prg   4   "change disk"       prg
8    "bug1/3/008"       seq  8   "bug1/5/015"   seq   4   "bug1/6/018"    seq   11  "c64 datamv.pal"      prg   4   "check disk"        prg
11   "bug1/3/009"       seq  10  "bug1/5/014"   seq   8   "bug1/6/017"    seq   1   " more emulation "    prg   4   "key"               prg
12   "bug1/3/004"       seq  12  "bug1/5/004"   seq   10  "bug1/6/011"    seq   2   "irqx"                prg   9   "menu"              prg
9    "bug1/3/005"       seq  10  "bug1/5/005"   seq   10  "bug1/6/012"    seq   18  "quadra basic"        prg   6   "renumber"          prg
10   "bug1/3/cov"       seq  10  "bug1/5/007.1" seq   8   "bug1/6/013"    seq   2   "quadra basic ml"     prg   2   "address list"      seq
12   "bug1/3/002"       seq  9   "bug1/5/012"   seq   13  "bug1/6/014"    seq   16  "editor"              prg   1   "renum.test"        prg
11   "bug1/3/011"       seq  7   "bug1/5/013"   seq   13  "bug1/6/015"    seq   35  "asmbler"             prg   1   "renum.1"           prg
12   "bug1/3/012"       seq  10  "bug1/5/014.1" seq   15  "bug1/7/cov"    seq   5   "demo-ml.sou"         seq   6   "renmbered"         prg
14   "bug1/3/003"       seq  8   "bug1/5/011"   seq   12  "bug1/1/cov"    seq   13  "unassembler.note"    prg   344 blocks free.
14   "bug1/3/010"       seq  6   "bug1/5/007"   seq   12  "bug1/1/002"    seq   36  "unassembler.kern"    prg
9    "bug1/3/015"       seq  16  "bug1/6/cov"   seq   8   "bug1/1/004"    seq   38  "unassembler.bas"     prg
5    "bug1/3/013"       seq  13  "bug1/6/002"   seq   8   "bug1/1/005"    seq   36  "unassembler.mae"     prg
```

CBUG UTILITIES & MISC #2

CBUG #M20 NEW RELEASE #12768

Many members who have upgraded their B128's to 256K found that the 8432 emulator as first published would operate a B128 in only 2 banks, even though there are 4 available. Now with 1Meg, the fixing of the 8432 emulator was manditory -- so here it is. This upgrade will allow use of nearly all 15 banks available in a 1Meg B128, simply by using SHIFT COMMODORE key, bank #. If you thought you bought just one computer, Bunkey, look again. Even with an off the shelf B128, you can have it think it is TWO 8032 computers, with 256K, FOUR 8032's, etc. Each 8032 can be running a different program better yet! Now the emulator is not a completely perfect "cat's meow". I will not run all 8032 programs, particularly those requiring special hardware, program ROMS, having clock calls or a few other illegals. None-the-less, lots of opportunity.

Mr. O'Boyle has profered an extensive suite of electronics formulas. Mr. Rotello's 8050 speed up and reset are on the disk. I've tossed in an example of Superscript templates for completing and calculating bank deposits, and a standard letterhead template along with some comments on Superscript efficiency thru pre-forms. Mr. Loeffler has provided tow extensive essays on the bugs in the CABS/INFO designs accounting suite, together with the solutions he has devised for many of them. Mr. Bennett gives is some custom programming in the field of statistics.

NORTHWEST MUSIC CENTER INC
107 S. NORTHWEST HWY
PARK RIDGE, IL 60068
312-823-9605

DON'T MISS OUT ON THESE TREMENDOUS BARGAINS

We have bought a huge shipment of Commodore B-128 software and peripherals. We are offering these programs and hardware at the following unbelievable prices. Note there are limited quantities of most of these items.

PRECISION SOFTWARE

Superbase	sealed	$29.95
	opened	$19.95

ONE OF THE FINEST DATABASES!

Superscript II	sealed	$29.95
	opened	$19.95

C.A.B.S. ACCOUNTING

General Ledger	$29.95
Accounts Receivable	$29.95
Accounts Payable	$29.95
Order Entry	$29.95
Payroll	$29.95

Note these programs run only on the B-128.

HANDIC SOFTWARE

	list	price
Calc Result	$195	$99
Word Result	$195	$99

Note:Word Result integrates with Calc Result. Memory must be at least 256k. With the integrated version you can pull calculations out of the spreadsheet and insert them into the document. You can also switch back and forth between them. A simple database is included for mail merge and other simple uses.

NWM ACCOUNTING

* Inventory $50
w/ built in calculator
uses name or part no. as key

Available March 1st, 1986

* Point of Sale $50
Integrates with above and allows for real time invoice printing without disk swaps.

Available April 1st, 1986

* Note: applications require the use of Superbase to run.

B-128 Programmers Reference Guide listed for $29.95 now only $14.95

	List	Price
P to I cables	$75	$25
Pet switches	$295	$199

run multiple keyboards on 1 or more drives and printers.

We also have refurbished COMPUTERS, DRIVES and PRINTERS for low,low prices. Call for details! THESE COULD BE THE LAST OF THE B'series

SINCE ARE ALSO A MUSIC STORE WE HAVE HUNDREDS OF MUSIC ITEMS AVAILABLE. MESA BOOGIE AMPS,GUITARS,KEYBOARDS,ACCESSORIES! CALL FOR DETAILS! HUGE DISCOUNTS AVAILABLE!

CALL BETWEEN 1PM and 8PM CST
HURRY QUANTITIES ARE LIMITED!

Prices FOB Park Ridge, Illinois

CBUG ADVERTISEMENTS

PHYSICAL EXAM
MIRACLE DISK ALIGNMENT PROGRAM

This is a MUST HAVE program. Even if you are sure your drives are running correctly, you really don't know. Just because they appear not to cause trouble, they could be marginal -- and one of these days you will be looking at corrupted and unrecoverable data. It's not good enough just to drive somewhere on the road, you have to be in the right lane at the right speed. Don't risk a data crash -- keep an eye on your drive performance!!!

Now that we can surgically correct some, and soon all of the drive speed problems, no doubt most everyone will want to make the upgrades. BUT, before undertaking the upgrade, you MUST have an alignment method to get things running after surgery. Physical Exam is IT!

About 98% of drive and data loss problems are easily traceable to speed errors. The majority of the members calling me with problems are because of this single flaw. Don't be caught with your pants down. Order yours yesterday.

ORDER PART #12219 for the 8050
12223 for the 1541 Only $35.00 each.
12238 for the 4040

> WARNING WARNING WARNING
>
> Do NOT attempt to exchange drives on the MPI versions of the 8050's as per para. 4, pg. 13 without obtaining pertinent errata sheets from Cardinal at the address in your instruction book.
> You may seriously damage your drive. (It has just been discovered by a member that there are several previously unknown versions of of the MPI drive configurations manufactured by CBM in common use by CBUG members.
> This is an important example of why to send in registration cards-- as those who registered have already received their errata sheets!

CBUG ADVERTISEMENTS

BETTER ZENERS and REFERENCE DIODES

Finding parts is not always so easy, so CBUG has purchased a supply of both replacement close tollerance zeners of the type recommended by Mr. Kernaghan, and a far superior reference diode. We are offering these, albeit at a modest markup, along with copies of the component locations on circuit boards and with schematics and brief instructions. KNOW WHAT YOU'RE DOING OR GET SOMEONE WHO DOES TO HELP!

Zener Diode: ECG 5014A is also known as a SK6A8 under the RCA numbering system and is in reality a relabled 1N5235B. These diodes can simply be swaped for the two located on the analogue board, one each for drive 0 and 1. They are about 4 times better than the original part ratings. These parts are rated at .05% per degree C drift. Two are provided in a package. Order #11330, $6.00

Precision Reference: LM329B is a tight tollerance near zero drift reference diode from National Semiconductor. It sports a .0015% per degree C drift -- 33 times better than the zener diode. However, a bias resistor must also be changed, so 4 parts need to be substituted. Make twice as sure you are capabable of this before attempting. This upgrade will be necessary when adding Gerry Mills Phase Lock Loop as well. Two LM329's and 2 2.7K resistors in a package. Order #11344, $8.50

CBUG ADVERTISEMENTS

SUPER DISK DEAL

5.25'' SSDD $.85 ea

Probably the finest and most advanced disks available today. 100% certified. Buy back warrantee! Opus (brand) manufactures the magnetic media itself, rather than buying other companies' seconds and just packaging the disks. Total quality control and proprietary formulas assure the near zero failure and the extremely long life rates for which OPUS computer tape -- and now floppy disks are now nationally recognized. Disks are polished to a high gloss for maximum head life. The second largest BBS in Chicago is now using Opus disks and reports that where all others (40+ brands tried) last at the most 2 to 3 days, the Opus disks have yet to fail after many weeks. The Green Machine BBS handles 70,000 calls per month, and the disk drives log 9 to 11 hours actual rotation time per day!. We've sold over 20,000 to critical business application users (IBM, CBM and others) without a single return!

Disks are soft sectored, w/ hub rings, extra heavy weight lined jacket. Packaged in 10 count poly bags, retail labeled, sleeves, plus packet of tabs and labels -- all but the retail box and shrink wrap! A full line of products is available including Double Sided, Retail Boxed, unlabeled bulk, 3.5", 8", specially formated disks and storage boxes. Contract pricing available.

Order using the CBUG order form inside the front cover of this publication. For complete price list and order forms write CBUG.

PAID ADVERTISEMENTS

KNIGHTS COPY UTILITY

This is the one you've all been waiting for, and about 1/3 of CBUG members have. This lovely program will copy anything presently known to date for use on the B128 (except itself) -- INCLUDING Superscript and Superbase!!! It takes precisely 10 minutes to run, and will make copy after copy simply by shoving more blank disks into drive 1 of your 8050! If you wish to change originals, you switch the disk in drive 0 and continue. When you first boot this program it loads into the 8050, NOT the computer. In 10 seconds you can disconnect your IEEE cable from the disk drive and the 8050 will run all by itself. You can use your computer for some other purpose.

We use this program to duplicate our library disks -- and it works great. Once this program is loaded, the three indicators flash like traffic lights, till the copy process starts. If there is an error, the 8050 stops and the green light turns red and does not flash. Super! Not for real old drives -- DOS 2.7 only.

Now folks, you all see these fearsom copyright notices on programs. Yes, it is illegal to make copies for others, but the U.S. Congress says it is OK to make backup copies for your own use. This is known as the Fair Use Doctrine.

Don't risk ruining your Superscript or Superbase disks. Back up everything you have without having to press a single button!

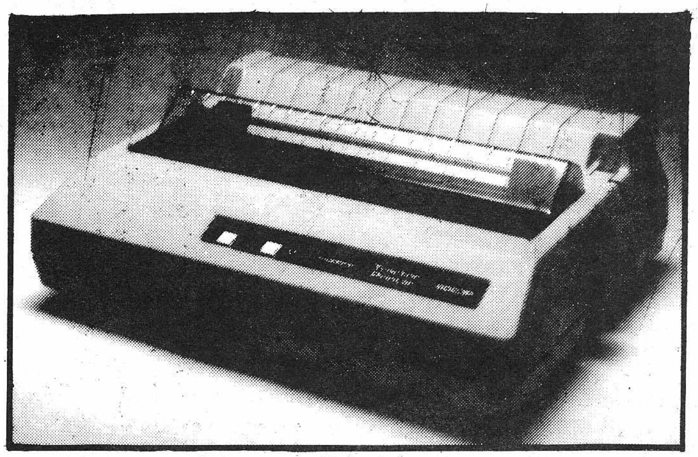
Order from CBUG -- stock #12204 only $20.00 ea.

PROGRESSIVE PERIPHERALS & SOFTWARE

464 Kalamath Street
Denver, Colorado 80204
300/825-4144

Precision Software

"Your One-Stop Commodore Shop"
WE SUPPORT B-128 USERS!!



High Speed 15½" Wide Carriage 8023P IEEE Printer
The BEST printer for your B-128!

It's fully compatible with word processing, database and other business and personal software.

Here are some of the 8023P's great features:

- 15½", 136 to 250 column carriage
- 150 characters per second!
- Full Commodore graphic/reverse characters
- Tractor or friction paper feed
- TRUE decenders!
- 1 MILLION character cartridge ribbon
- 100 MILLION character print head life!
- Built in diagnostics and self test
- Handles thick three part forms
- COMPLETE 50 page user manual
- Condensed and Near Letter Quality modes
- Completely programmable print format!

ORDER TODAY! GET IN ON THIS GREAT DEAL!

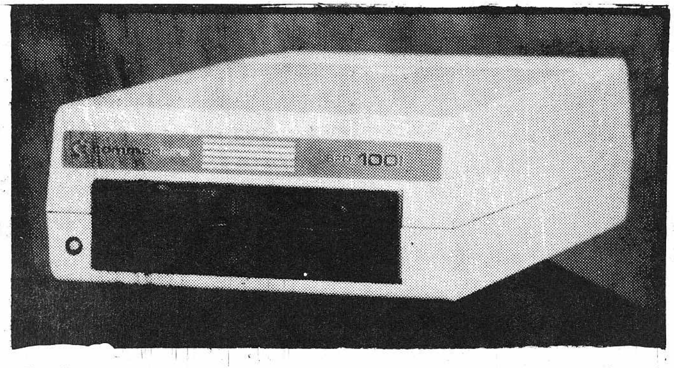
LOW AD PRICE
ONLY $179.95

FREE!!

EXTRA RIBBON WITH EACH PRINTER!!!

ONE MEGABYTE IEEE DRIVE

The Commodore SFD-1001 stores over one megabyte of data on a double sided double density disk. Fast IEEE interface is plug-in compatible with a B-128 or PET computer. The SFD-1001 has the same format as an 8250 and will load 8050 and 8250 software without conversion! The best feature? The built-in price of only $239.95!



SERIAL IEEE INTERFACE FOR THE C-64 & C-128

The E-Link is a 100% transparent interface, connecting IEEE peripherals to the serial port of your computer!

E-Link works with 4040, SFD-1001, 8050, 8250 disk drives, and all other CBM IEEE peripherals. Plug the E-Link right into the serial port of your computer, no dip switch or software settings to adjust. Takes no power away from your computer and runs at almost twice the speed as normal serial devices. Reg. $119.95 NOW only $99.95!!

LYNKER MODEM - 300/1200 BPS

Introducing the Lynker . . . that's right, our new 300/1200 BPS Modem! Just plug it into your computer's serial port, and you are ready to transmit and receive messages.

The LYNKER is 100% Hayes compatible and uses all Hayes commands. An independent power supply eliminates power problems associated with many other modems that rely on the CBM power supply. Eight LED indicators tell just what the modem is doing at any time.

- High speed
- Carrier detect
- Receive data
- Modem ready
- Answer mode
- Off hook
- Send data
- On/Off
- Speaker
- Hardware Dip Switch controllable

RS 232

The LYNKER is completely Bell 212A/103 compatible. The independent power supply is U.L. and C.S.A. approved.

The LYNKER is comparable to the Hayes Smartmodem 1200. The Smartmodem retails nationally for $369.95 . . . The LYNKER is now available to you for only $199.95!

Cables available.

ORDER AN SFD-1001 OR 8023P & GET $5.00 OFF THE PRICE OF A CABLE OF YOUR CHOICE!!!

NEW SOFTWARE FOR THE B-128!!

SUPER DISKDOC UTILITY: FIX CORRUPTED DISKS!

Suitable for any Commodore computer/disk drive combination, this utility set allows you to zoom in on the bytes on your disk, interpret them in plain English, hexadecimal or ascii, make any changes and replace them. You can detect the cause of frustrating disk read and write errors, and recover previously unreadable files and large portions of apparently deleted data. You can either copy any part of any file, change directory entries entirely, or verify the integrity of internal disk pointers and links.

Super Diskdoc come with a simple menu interface. You can type the command or cursor along to it. Once you call up the file or the part of the disk you want to look at, you can browse in any direction through it. Clear documentation is included for ease of use.

Price $25.00

Superscript *Superbase*

CALL TODAY TO ORDER!!! (303)825-4144

SUPERSCRIPT III: IMPROVED B-128 WORD PROCESSING

Most of the following features will be familiar to users who have purchased Superscript for the C-64 or C-128, but will be new to the B-128 users:

- Friendly "duck-shoot" (Lotus-style) menu interface
- User-defined command sequences (macros)
- User-defined printer interfaces
- Two text areas with the ability to move data from one to the other
- Better integrated spelling checker
- Improved mailmerge, search & replace and calculator
- Multi-level indentation
- Directory selection of file names
- Single-drive back-up
- Keyboard bounce problem fix
- Disk-knocking protection removed

NEW!!!

Ugrade price **$45.00**
New Price .... **WAS $79.95 NOW $60.00**

ONLY WITH THIS SPECIAL ORDER FORM!

SUPERBASE VERSION 2: Upgrade NOW!

The following features will be new to B-128 users:

- Faster file definition load and program
- "Disk full" protection
- Improved lables program
- Database copy and recovery
- Delete utility
- MANY BUGS FIXED
- Disk-Knocking protection removed

NEW!!!

Upgrade price **$25.00**
New Price .... **WAS $79.95 NOW $60.00**

AT LAST . . . SUPERBASE: THE BOOK

The essential in-depth reference work for Superbase users, it is also a valuable guide for anyone using a database. Whether you are just starting out with Superbase or you want to use Superbase's programming language more effectively, this book should be your companion. Designed for all levels of experience, it includes dozens of practical examples. A clear and informal style makes learning a pleasure. By the time you have finished you will be a Superbase expert.

Price .... **REG. $20.00 NOW $15.95**

ORDER NOW!!!

ORDER HERE!!

CHECK THE BOX(S) OF THE ITEM(S) THAT YOU WISH TO ORDER:

□	______	8023P PRINTER **Reg. $199.00**	**NOW ONLY $179.95**
□	______	LYNKER MODEM	each $199.95
□	______	SERIAL TO IEEE INTERFACE **Reg $119.95**	**NOW ONLY $99.95**
□	______	PET TO IEEE CABLE (For connecting your CBM computer to an IEEE Peripheral)	each $34.95
□	______	IEEE TO IEEE CABLE (For connecting one IEEE peripheral to another)	each $34.95
□	______	SUPER DISKDOC UTILITY	each $25.00
□	______	SUPERSCRIPT III **WAS $79.95**	**NOW $60.00**
□	______	SUPERSCRIPT UPGRADE**(With Manual)**	each $45.00
□	______	SUPERBASE VERSION 2 **WAS $79.95**	**NOW $60.00**
□	______	SUPERBASE UPGRADE	each $25.00
□	______	SUPERBASE:THE BOOK **WAS $20.00**	NOW $15.95
□	______	SUPER DISKDOC UTILITY	**A MUST AT $25.00**
□		**INFORMATION ABOUT OTHER PP&S PRODUCTS**	FREE

ALL PRICES GOOD WITH THIS SPECIAL ORDER FORM ONLY! (OR CALL IN REFERENCE TO IT). ORDERS FOR SUPERSCRIPT III AND SUPERBASE VERSION 2 MUST BE SENT WITH AN ORIGINAL DISK. NO EXCEPTIONS!!

NAME ______________________________

STREET ADDRESS ______________________________

CITY/STATE/ZIP ______________________________

PHONE (   ) ______________________________

SIGNATURE ______________________________

MAKE CHECKS PAYABLE TO "PROGRESSIVE PERIPHERALS & SOFTWARE".

ORDERS FOR UPGRADE DISKS (SUPERSCRIPT III, SUPERBASE V. 2) MUST BE SENT WITH AN ORIGINAL DISK FOR EXCHANGE.

CARD NO. □□□□□□□□□□□□□□□□

EXP. DATE □□□□

□ **VISA**
□ **MASTERCARD**
□ **AMERICAN EXPRESS**

□ I ENCLOSE MY MONEY ORDER/CASHIER'S CHECK FOR $ ________

□ ENCLOSED IS MY PERSONAL/COMPANY CHECK FOR $ ________
I UNDERSTAND A 2 WEEK HOLD WILL BE PUT ON MY ORDER

□ PLEASE BILL MY VISA OR MASTERCARD AT THE TIME MY ORDER IS SHIPPED

CREDIT CARD ORDERS ADD A 5% SERVICE CHARGE TO AMOUNT DUE.

PLEASE INCLUDE AREA CODE. CREDIT CARD ORDERS MUST HAVE A PHONE NUMBER. ALL ORDERS INSIDE THE U.S.A. WILL BE SHIPPED VIA UPS. PLEASE DO NOT USE A P.O. BOX ADDRESS. ALL ITEMS ARE IN STOCK AND WILL BE SHIPPED PROMPTLY. FOR SHIPPING CHARGES, ADD $10 FOR EA. SFD-1001 ORDERED, EVERY 8023P ADD $17, AND EVERY E-LINK $2. THESE CHARGES INCLUDE INSURANCE AND HANDLING FEES. FOR ORDERS IN CANADA, SHIPPING CHARGES ARE DOUBLE THE U.S.A. RATE. ALL ORDERS MUST BE PAID IN U.S. FUNDS. PLEASE DO NOT SEND CASH!

SEND YOUR COMPLETE ORDER FORM TO:

PROGRESSIVE
PERIPHERALS
& SOFTWARE

464 Kalamath Street
Denver, Colorado 80204
303/825-4144

ALL PRICES EXPIRE MAY 31, 1986

Commodore 64, Commodore 128, CBM 8250, SFD-1001, and CBM 8023P are trademarks of Commodore Business Machines, Inc.

PAID ADVERTISEMENTS

EAST FLORIDA PLANNING AND DEVELOPMENT COMPANY
COMMODORE B - 128 / CBM 8032 PRODUCTS
PRICE LIST AS OF JANUARY 1986

Product Code	Name/Description	Price
PDS - 1	Commodore Public Domain Series Math I *	10.00
PDS - 2	Commodore Public Domain Series English I *	10.00
PDS - 3	Commodore Public Domain Series English II *	10.00
PDS - 4	Commodore Public Domain Series History I *	10.00
PDS - 5	Commodore Public Domain Series Games I *	10.00
PDS - 6	Commodore Public Domain Series Games II *	10.00
PDS - 7	Space Games - Trek2/Flight simulator/B-1/more *	10.00
PDS - 8	Electronics - Ohm's law/Reactance/Pwr. Suppply/more *	10.00
PDS - 9	Programming/Disk cracking utiltites	10.00
PDS - 10	Fancy Screens - Screen Ed./Screen Printer/Banner/more	10.00
PDS - 11	Cal. BUG Utilities Disk # 2 (June '85 version)	10.00
PDS - 12	Basic Program Conversion Utilities Disk (inc Xmodem)	10.00
PDS - 13	More Games - Adventure/Elisa/Poker/more *	10.00
ECS - 1	Home Management - Checkbook/coupons/budget/energy savings/ appointment book/more	20.00
ECS - 2	Adult Stimulation - adult party game (X RATED)	20.00
ECS - 3	Ham Radio - Morse code/Ant. desg./ARRL awards/Sat. tracking/ more *	15.00
ECS - 4	Flexfile - large data base program *	15.00
ECS - 5	CONFORM - Construction administration package	100.00
ECS - 6	Construction estimating (single family homes)	100.00
ESC - 7	Feasibility Studies - warehouse/office buildings/subdivisions/ convenient stores/more	150.00
ESC - 8	Pascal compiler *	50.00
ECS - 9	New Games - Spelagame/warehouse/phrase game/more	20.00

Other PDS and ESC disks available send $ 1.00 for list (free with order)
* code indicates available for the CBM 8032
Backup copies of any disk, inc. copywrited software available for $ 5.00 with proof of original ownership.

HARDWARE PRODUCTS

EHP - 1	6 ft. IEEE-488 male/male cable (non-piggy backed)	15.00
EHP - 2	Printer select switch (1 in/2 out, all female)	55.00

ORDERING INFORMATION

Please send the following to: E F P & D
19 Fairway Drive
Cocoa Beach, FL 32931

1. Your name and address (No PO Box nos., please)
2. Product code, number of units and machine of use
3. Disk drive format (8050, 4040, 8250, etc.)
4. Total price inc. sales tax if you live in Florida
5. Money order payable to E F P & D (sorry no checks)

We pay all shipping costs. All orders shipped in one week if in stock. Allow maximum of 4 weeks for delivery if not in stock.

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ PAID ADVERTISEMENTS ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

0	
1	**In a native B128, you have two banks, #1 for programming and;**
2	**#2 for data/variables.**
3	**A B256 or a B128 with an additional 128K added has banks #3 for data;**
4	**and bank 4 for more data, etc.**
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

The rest goes begging. Available now for the new low price of $350.00 factory installed and UPS insured return freight paid, you can move up to the full 1MEG. Have up to 9 banks in Superscript, 512K in Calc Result, 12 8032's in the 8432 emulator, and who knows what else our clever programmers in CBUG will do with all this capacity. Disk RAM? Multiple programs "running" at the same time...?

This technique uses no additional power, adds no heat. Is entirely installed on the existing computer mother board.

AVAILABLE NOW!!!! From Fred King, King Communications, 1804 Plover Spring Drive,
Plover, Wi. 54467. Call for scheduling 715 341 1149 6pm to 10:30pm CST.

FOR SALE

4023 printer ' working condition. $75.00 or best offer. Also a Xerox 1340 hytype 1 model 1200 teleterminal. Interface problem- Best offer. Mark Schwarzbauer- 1723 Michigan Sturgeon Bay WI 54235- 414-743-4151

- - - - - - - - - - - - - - - -

MOTHERS COVERS - They keep your B128 and 8050 cozy at night. Strong vinyl material, custom fitted to look sharp and protect your equipment. Brown in color. Both for $18 plus $1 for postage and handling. Check or money order to Mothers Covers, 2089 Agate St., Orange, Ca. 92667

- - - - - - - - - - - - - - - -

Extra B128 system, B128, 8050, 4023, amber monitor & 5 utility disks. $500 plus shipping. Jim Rhodes. 900 S. Dickenson Ave., Sterling, Va. 22170, 703-435-8121 evenings.

- - - - - - - - - - - - - - - -

B128, 8050,4023,monitor, SuperBase, Superscript. $700.00. Harold Bonella, 4366 Alice Way, West Valley City, Utah 84119, 801-969-3988

- - - - - - - - - - - - - - - -

B128, 4023, amber monitor, cassette recorder, cables, Programmer's Reference Guide, Superscript. $500, John Connell, Box 344, Warren Vt. 05674, 802 496 5546.

- - - - - - - - - - - - - - - -

B128 FUNCTION KEY OVERLAYS: Keep track of your custom programmed function keys. Blank overlays, 140# stock ruled on both sides to provide 2 sets of functions for each overlay. First Class postage paid. 3 for $2.00 - 4 for $3.50. Iowa residents add 4% tax. Cash, check or money order to: OFFICE ASSISTANCE - 4218 PINE VIEW DR. N.E. - CEDAR RAPIDS, IA 52402. Pre-printed overlays for quantity users available. Advise details for quote.

- - - - - - - - - - - - - - - -

AGRICULTURAL SPREADSHEET TEMPLATES FOR THE B128 FOR SALE: I have about 30 spreadsheet templates encompassing crop and livestock production, depreciation, finance, and etc. in VISICALC format which may be readily converted to CALC-RESULT format. COST: $15 to $25 each depending on the complexity. David Johnson, JFE COMPUTERS, RFD #2 Box 262, Genesco, Il. 61254

- - - - - - - - - - - - - - - -

HOT TIP from CBUG: Once upon a time CBM offered a superlative letter quality printer, model 6400. This was a C-ITOH F10-40 aka Diablo 630. It is a 40 character per second daisey wheel with 15" capacity, optional page and tractor feed. It is software (directly from Superscript) addressable for 10, 12, and 15 characters per inch, 6, 8 and 12 lines per inch; full micro justification as in proportional spacing type setting, bi-directional platten for plotting, etc. The 6400's have been re-modified as Centronics ported (you'll need an IEEE to Centronics adaptor from CSI -- see ad in last CBUG ESCAPE, (305) 725 6574) --There are two liquidators handling this item: JS&A of Northbrook Il. 800-228-5000 @ $499.00, and BCE Liquidators, Washington D.C., 800-545-7447 @ $349.00 -- ask for Tom McGuire. Neither company has elected to place an ad in the CBUG ESCAPE. This information is offered courtesy of CBUG.

- - - - - - - - - - - - - - - -

ANOTHER TIP from CBUG: Computer Specialties (305-725-6574) has reduced the price on the ST10C 10 Megabyte hard drive to $995.00 factory direct. Add $50.00 each for Hard Disk versions of Superbase and Superscript (will integrate).

WANTED TO BUY

8050 drives, working or not. Describe condition, include phone number and asking price inletter to Mushroom, the Journal of Wild Mushrooming, Box 3156, Moscow, Idaho 83843.

- - - - - - - - - - - - - - - -

Main board for 8050 drive (Micropolis drives). A. Staub, 2230 E. 4th Ave, Denver, Co. 80206, 303-355-0018

- - - - - - - - - - - - - - - -

8050 drive in good condition. Don R. Moses, Attorney at Law, P.O. Box 67, Jellico, Tennessee 37762, 615-784-8290

- - - - - - - - - - - - - - - -

PRELIMINARY ANNOUNCEMENT

AN ASSEMBLER WRITTEN SPECIFICALLY FOR THE B128!!!

By: Howard Harrison

Assembler 5.5 is a powerful editor/assembler for your B-128 (requires at least 8k of additional memory in bank 15)
Availability: must be purchased through CBUG.
When: will be available before the next publication ESCAPE.
Cost: $35.00
By: Howard Harrison.

Editor features:
Full screen scrolling editor with single keystroke commands.
Tokenized file usage for memory efficiency (about 1/2 of normal storage)
Tokenizing can be turned off with 2 keystrokes!
Hunt, Goto, & Rehunt commands. Two user definable text buffers
Built in DOS wedge with 4-column disk directory. multiple drive support.
May print all or any part of the text in memory, also go to any printer device, or even disk!

Assembler features:
Extremely fast assembly time because everything can be done in memory, without using the slow disk!
If 64K is not enough for your source, you can assemble source directly from DISK (just use memory for labels), or use library files for those common routines.
The assembler is capable of generating multiple object programs in one assembly. This is a life-saver for people who code in multiple banks.
Generates relocatable object code.

The editor and assembler lives in 8k ram, which can be anywhere of bank 15. The start and end of the source buffers can be in any bank. The default disk & printer devices can be changed.
The symbol table is stored in sorted order for a fast binary search during assembly. Symbol table listings can go to printer or disk with up 9 definitions on a line.

Comming soon: debug 5.5, an interactive assembly language debugger for assembler 5.5.

Now here's a useful example, easily adapted to most anything. Mr. Enochs is a Savings & Loan Appraiser. You can well guess the quantities of forms he has to fill out for each appraisal. We he put it on Superscript, and here are the templates, instructions, the whole works! Lots of work and experimentation for him -- for our members, mere copywork! From Mr. Caple is a program for calculating time cards -- as in payroll. And if you are not game for a major spread sheet program to do your small home accounting chores -- thank Mr. Peterson, for he has done that for your too, again in Superscript.

And if you want a chuckle, our libraian Mark Schwarzbauer sat down and wrote a cartoon strip that runs on the B128 -- just dload"cartoons" and type RUN --

Lastly, from our Delphi SYSOP are the main HELP files from the Delphi Forum to save you from having to down load them uefore you know how to do so!

```
1 "CBUG Utilities 2" 76 2c      create input         2    "file.translate"  prg  10  "correlation"      prg  6    "janbkdocrec86+"  seq
8432 upgrate for 256 & 1Meg     create instr         8    "com.emit.bin"    prg  2   "-----ge-----"     seq  17   "explainlet1+"    seq
4    "Start8432.21"     prg     create output        4    "zener.reg.bin"   prg  12  "appform.instr."   seq  2    "-----ms-----"    seq
2    "t1"               prg     dar.reg.bin          8    "std.reg.bin"     prg  59  " <b>BROUGH: Rob"  seq  4    "cartoons"        prg
2    "td.alt"           prg     day from date        1    "capacitor table" prg  77  "appform.ss2+"     seq  8    " cartoon1"       prg
2    "t2"               prg     depa                 8    "dar.reg.bin"     prg  12  "pre-app"          prg  8    " cartoon10"      prg
2    "td"               prg     depb                 13   "reg.design"      prg  4   "merger"           prg  8    " cartoon11"      prg
81   "8432.21"          prg     depc                 1    "resistor table"  prg  25  "home budget"      prg  8    " cartoon12"      prg
4    "bank F.20  '1024" prg     editor               1    "-----jr-----"    seq  1   "app.print.basic"  prg  8    " cartoon13"      prg
4    "instructions8432" prg     fft                  1    "q" 8050 speed up prg  7   "disk instr"       seq  8    " cartoon14"      prg
1    "---above---"      seq     file.translate       1    "r"  " reset      prg  3   "sys finder"       prg  8    " cartoon15"      prg
2    "----j0'b----"     seq     h.s. analysis        4    "-----nd-----"    seq  57  "appform.ff+"      seq  8    " cartoon16"      prg
3    "  directory  "    seq     hex poke             5    "5+" deposit temp seq  22  "name/add.ff"      seq  8    " cartoon17"      prg
     ac.form                    hist                 4    "6+"  "     "     seq  8   "appbill.ff+"      seq  8    " cartoon18"      prg
     act.filt                   least squares        1    "66+"letter   "   seq  1   "-----cc-----"     seq  8    " cartoon19"      prg
     amort                      logic simulation     3    "-----bl-----"    seq  1   "FROM"             seq  8    " cartoon2"       prg
     asmbler                    matrix.determ        118  "cabs i/m bugs"   seq  15  "timecard"         prg  8    " cartoon3"       prg
     bessel.func.1              max                  48   "cabs defect/bitc" seq 1   "-----fp-----"     seq  8    " cartoon4"       prg
     bessel.func.2              mem manipulator      2    "-----rb-----"    seq   Spread sheet template in SS 8    " cartoon5"       prg
     capacitor table            mem.util             25   "STATISTICS"      prg  4   "jandoc86r+"       seq  8    " cartoon6"       prg
     capacitynetworks           p.s.design           12   "distribution"    prg  5   "jandoc86d+"       seq  8    " cartoon7"       prg
     check disk.rev1      5   "Cat.bin"       prg    4    "sample size"     prg  5   "jandoc86pc+"      seq  8    " cartoon8"       prg
     com emitter amp      4   "h.s. analysis" prg    7    "confidence intv" prg  5   "sumdoc86r+"       seq  8    " cartoon9"       prg
     com.emit.bin         3   "disk.DOCS"     seq    9    "chi square"      prg  4   "sumdoc86d+"       seq  154  "Delphi help 1    seq
     compact              2   "hex poke"      prg    11   "t test (i)"      prg  5   "sumdoc86pc+"      seq  154  "Delphi help 2    seq
     compact.c            9   "mem.util"      prg    10   "t test (d)"      prg  5   "sumdoc86t+"       seq  267 blocks free.
```

RETAIL NEWS DISTRIBUTION CBUG #21 NEW RELEASE #12699

Mr. Freeman is in the retail news distribution business in Newtown Square Pa. In recent years Mr. Wes Robinson developed this suite for managing a route oriented distribution business as a contract programmer to Mr. Freeman. The programs will run in both the B128 or the 8032. More interestingly, with the 8432 emulator you can run multiple segments simultaneously! Documentation is in the "ss find 1/86" file to be read using Superscript. These programs are copyrighted by Mr. Robinson and any distribution beyond CBUG is expressly prohibited.

```
1 "find prg disk  " 30 2c  11  "do count"          prg  5    "routelist"         prg  1   "n belfield"   seq  6    "59"              seq
19   "main menu"     prg   6   "dear customer"     prg  11   "so count"          prg  1   "bon air"      seq  13   "60"              seq
31   "ss find 1/86"  seq   10  "inq list"          prg  3    "spin disk"         prg  1   "bon air2"     seq  13   "61"              seq
116  "bill menu"     prg   8   "1 listd1 abbr"     prg  8    "file concat"       prg  1   "bon air3"     seq  13   "62"              seq
116  "sun bill"      prg   2   "list autorun"      prg  2    "screen print"      prg  1   "brierwood"    seq  15   "63"              seq
112  "data load"     prg   2   "list newsmaster"   prg  119  "bill b128"         prg  1   "ablocklist"   seq  5    "menu b128"       prg
19   "block menu"    prg   3   "load autorun"      prg  15   "loadtrans instr"   prg  1   "atest 1"      seq  3    "anoso-11/18"     seq
5    "auto1"         prg   4   "load newsmaster"   prg  9    "pdl"               prg  1   "atest 2"      seq  3    "ss end of find"  seq
9    "auto3"         prg   6   "name search df"    prg  8    "righteven"         prg  1   "anoso"        seq  1    "ss---misc---"    seq
8    "auto3sun"      prg   2   "newsmaster"        seq  8    "lefteven"          prg  14  "38"           seq  7    "labels6.p"       seq
3    "bill notice"   prg   12  "pay rec"           prg  1    "annabella"         seq  9   "92"           seq  1    "----from----"    seq
8    "code 7"        prg   12  "pay rec so"        prg  1    "ashton"            seq  16  "57"           seq  1186 blocks free.
7    "code 4"        prg   2   "pay rec lines"     prg  1    "ashton2"           seq  14  "58"           seq
```

MATH EDUCTATION PROGRAMS CBUG #22 NEW RELEASE #12701

This is a series of math education programs developed by Bill Townsend over the last two years which were presented last fall at the Indiana Computer Educators Comference. They print out worksheets and tests for high school math classes. Most of the programs are for the Refresher or General Math Classes, but there are a few for eighth grade math and a few for the higher math class.

Mr. Townsend has kindly donated this series to the CBUG membership for its use. Others need secure permission from Mr. Townsend.

```
1 "math programs   " 54 2c  7  "interest prob"     prg  3  "print circle equ"  prg  8  "tax rate prob"      prg  4    "dec div 1361"      prg
8    "subt fractions"   prg 12 "range-med-aver"    prg  7  "alg equ +/-"       prg  7  "carpet cost"        prg  8    "bigfract mult"     prg
4    "assign computer"  prg 5  "averages"          prg  4  "subtraction 3"     prg  6  "solve circles"      prg  4    "simp salary"       prg
8    "subt big fract"   prg 3  "medians"           prg  4  "subtraction"       prg  7  "solve 2 circles"    prg  4    "gross salary"      prg
4    "comp pairs assig" prg 8  "trapezoid area"    prg  4  "subtraction 2"     prg  9  "pl figure area"     prg  5    "gross sal 1361"    prg
4    "computer assign"  prg 8  "rect solid ta&vo"  prg  4  "mult 3by2 digits"  prg  8  "solids vol/area"    prg  4    "fract of 1361"     prg
7    "add problems"     prg 8  "tri area sas"      prg  4  "large mult 3x3"    prg  4  "range/median/ave"   prg  9    "frac div 1361"     prg
6    "fifty add prob"   prg 9  "cyl la-ta-vol"     prg  4  "large mult 3x2"    prg  4  "multi average"      prg  9    "frac/div/1361"     prg
9    "subt 3 digits"    prg 9  "rect p& area"      prg  6  "large additions"   prg  4  "pythagoras -c"      prg  7    "frac redu 1361"    prg
6    "two digit div"    prg 8  "triangle area"     prg  4  "large div"         prg  5  "pythagoras -b"      prg  4    "dec/line/add1361"  prg
8    "two digit mult"   prg 8  "cone volume"       prg  7  "fract reduce"      prg  4  "tennis ball prob"   prg  4    "dec 60 line1361"   prg
2    "round cents"      prg 8  "sphere area/vol"   prg  4  "decimal line add"  prg  4  "ten/ba expt 500"    prg  6    "dec/mult/round"    prg
4    "line dec add"     prg 7  "mean variation"    prg  4  "large div r"       prg  4  "tri area (sss)"     prg  7    "simp int 1361"     prg
4    "metric sheet"     prg 6  "med/ran/ave/7"     prg  4  "big div rem"       prg  3  "tri area (sas)"     prg  4    "dec div-1361"      prg
4    "decimal mult"     prg 6  "ran-med-ave-8"     prg  4  "dec add line"      prg  6  "metric equals"      prg  4    "doll/cent round"   prg
5    "dec mult rd"      prg 9  "add/subt signed"   prg  8  "fract add big"     prg  6  "metrics 3x"         prg  4    "dec/part/of 1361"  prg
4    "division r"       prg 4  "mult signed numb"  prg  8  "big fract mult"    prg  5  "tri/area/sas"       prg  4    "dec of prob"       prg
4    "dec division"     prg 4  "division signed"   prg  5  "fract of $/ct"     prg  9  "pl fig area 2"      prg  4    "dec of calc 1361"  prg
2    "prime factors"    prg 2  "number lines"      prg  8  "fract div sp"      prg  9  "wk fed tax cal"     prg  4    "fraction of calc"  prg
3    "greatest factors" prg 6  "mult/div/signed"   prg  5  "fract of dollars"  prg  8  "la/vol solids"      prg  5    "comb pcts 1361"    prg
2    "fraction add"     prg 8  "graph equa 8"      prg  6  "calculator mult"   prg  1  "ans (25) sheet"     prg  4    "calc mult 1361"    prg
10   "trig triangle ws" prg 5  "circle cir/area"   prg  7  "calculator div"    prg  7  "sp add problems"    prg  4    "calc add 1361"     prg
10   "solve triangles"  prg 6  "rectangle p/area"  prg  1  "name flash"        prg  1  "pr let quality"     prg  6    "pct proportions"   prg
5    "fract mult/div"   prg 8  "parallelo area"    prg  7  "percent of 128"    prg  5  "mult 3x2 spec"      prg  13   "units sq ft etc"   prg
5    "memofractperct"   prg 4  "tri3sides area"    prg  4  "dec line add"      prg  1  "56 answer sheet"    prg  13   "sq units"          prg
4    "pertofraction"    prg 5  "tri2s/angle area"  prg  7  "calc mult/div"     prg  5  "3x2 mult spc"       prg  5    "scale mappings"    prg
5    "percent of"       prg 5  "rectangle solids"  prg  5  "subtraction sp"    prg  2  "25 ans sheet sp"    prg  10   "rect/peri/area"    prg
5    "perc of dec"      prg 5  "base/vol pyramid"  prg  5  "addition spec ed"  prg  4  "division two"       prg  11   "farm area prob"    prg
7    "trig prob"        prg 5  "ba/vol cylinder"   prg  5  "add spec ed 1"     prg  5  "7 number average"   prg  9    "area plane figs"   prg
7    "pect base"        prg 5  "base/volume cone"  prg  5  "mult spec ed"      prg  4  "dec/line/add 30"    prg  9    "la/ta/vo/cyl"      prg
5    "percentages"      prg 5  "ba/vol/la cyl"     prg  7  "feet-inch add"     prg  4  "decimal adds"       prg  10   "solids area/vol"   prg
5    "rate pct"         prg 4  "operations order"  prg  7  "ft-inch subt"      prg  4  "forty averages"     prg  9    "triangle 3 areas"  prg
7    "combined pct"     prg 6  "algebra equation"  prg  5  "48 additions"      prg  1  "ans (50) sheet"     prg  4    "FROM & letter"     seq
1    "ans sheets"       prg 6  "alg equ 4x"        prg  9  "doll/cent disc"    prg  3  "1361sp ans"         prg  1075 blocks free.
10   "discount prob"    prg 6  "hyperbolas print"  prg  9  "discount $/cents"  prg  5  "dec mult 40"        prg
```

BIBLE GAMES CBUG #23 NEW RELEASE #12735

Yet another item from our librarian, Rev. Mark Schwarzbauer. With a great deal of arm bending, Mark negotiated a fellow minister into allowing CBUG to publish these copyrighted programs for use by CBUG membership at a fraction of the usual $60.00 royalty applied in their releases for other machines. Great for the kids.

```
1 "R-SIG-2        " aa 2c   8    " instscrn"     prg   134  "luke"          prg   54   "farway"          prg   3    "the answers"     prg
1     " Startup"       prg  8    " bootscrn"     prg   141  "acts"          prg   1    "    directory  " seq   1550 blocks free.
6     " menu"          prg  97   "mark"          prg   24   "books"         prg   25   "answer key"      seq
```

8432 DISSASSEMBLED CBUG #24 NEW RELEASE #12720

Back last fall yours truly was wondering if the 8432 emulator might be modified to assist in running C-64 materials on the B128. To our good fortune two of the most experienced PET experts in the Fox Valley Commodore Users Group, Messrs Ken Magin and Paul Moch agreed to dig in and see what could be done. The first step, a huge one, was to dissassemble the 8432 emulator to study it. The project proved impractical, but they copyrighted their work and turned it over to CBUG for distribution amongst the membership lest all that work be of no value at all. So CBUG programming wizards, go to it!

```
1 "8432 disasbld   " 72 2c   56   "8432a-06.src"   prg   35   "8432.21-01"   seq   1    "8432.21.lib"      seq   48   "8432b.21-05.src"  prg
64   "8432a-01.src"     prg   45   "8432a-07.src"   prg   38   "8432.21-02"   seq   22   "8432.21.lab"      seq   31   "8432b.21-06.src"  prg
47   "8432a-02.src"     prg   22   "8432a-08.src"   prg   38   "8432.21-03"   seq   52   "8432b.21-01.src"  prg   1    "8432b.21.lib.src" prg
56   "8432a-03.src"     prg   1    "8432a.lib.src"  prg   34   "8432.21-04"   seq   55   "8432b.21-02.src"  prg   28   "8432b.21.lab.src" prg
49   "8432a-04.src"     prg   30   "8432a.lab.src"  prg   35   "8432.21-05"   seq   61   "8432b.21-03.src"  prg   4    "--- FROM (c) ---" seq
51   "8432a-05.src"     prg   34   "b"              prg   20   "8432.21-06"   seq   48   "8432b.21-04.src"  prg   1046 blocks free.
```

PRE RELEASE #4 PR #4 NEW RELEASE #12749

Lets see what we have in this potpourie. Just glossing over the authors, Garrett Bibb is one of our many capable programmers, as is Dick Wilkinson. Louis Lohman took a shot at a TPUG disk and found the only directly working program was "new list 8050" -- should give some others a challenge or two. Gary Anderson of the 1Meg memory upgrade gave us a screen action, bombs, trajectory and all. Jim Marusek wrote an extensive instruction file for Mr. Swan's Star Trek before there was one available and a couple of programs he's donated as well. Dave Runion provided a collection generally from TPUG including a simplistic disk speed display program -- no match for Physical Exam, but definitely better than tearing the drive apart to use the bottom strobe wheel! --plus a bunch of hand business and math programs. William Wolf addresses some of the issues in C-64 to B128 conversions and some hardware aids to help the project along. Jim Whatley provides us with programs that certainly belong on the beginning of the Educational Sig disks, in this case a series of programs for the typing teacher -- testing and much more. The grading programs are likely of use for other types of classes as well. Mather Goldstein sent in a collection of his favorites including a nice simple checkbook and tax record keeping program, and a direct to printer program so you can use your computer as a typwriter line at a time! And last, a new terminal program for the B128 from an editor of Transactor Magazine, Chris Zamara hand carried to us by Jessie Knight. This appears to be a machine language x-modem type program capabable of running all the way to 19,200 baud.

This, no doubt, is the best Pre-Release disk CBUG has ever offered. Nearly everything is expected to work right out of the box. The only reason we have put this stuff on a Pre Release disk is there is no place better to put it and get it all to you right hot bang now!
There's got to be something in here for everyone.

```
1 "Pre Release #4  " 74 2c   15   "cassette label.w" prg   42   "8050 update"       prg   43   "organic names.p"    prg   7    "grade averager"     prg
2    "-----gb-----"     seq   4    "checkbook.ss"     seq   3    "8050 menu"         prg   20   "rutherford.p"       prg   9    "superscript load"   prg
1    "-- utilities --"  seq   9    "checkbook.w"      prg   25   "8050 print"        prg   23   "density cal.p"      prg   7    "student grader"     prg
9    "read mo.BAS"      prg   6    "disk master.ss"   seq   25   "interest"          prg   12   "spd orbitals.p"     prg   15   "timed typing I"     prg
12   "to norman"        seq   31   "disk master v2.w" prg   23   "new list 8050"     prg   13   "scale reading.p"    prg   9    "disk information"   prg
16   "assembler.BAS"    prg   10   "diskmod.ss"       seq   4    "run me first"      prg   50   "data analysis.p"    prg   7    "add new program"    prg
6    "data entry.BAS"   prg   10   "diskmod.w"        prg   1    "-----ga-----"      seq   6    "consumer loan.8"    prg   2    "-----mg-----"       seq
3    "data correction." prg   9    "fkey template.ss" seq   29   "b128 bomber"       prg   9    "auto loan.8"        prg   13   " menu"              prg
13   "memory mon.BAS"   prg   28   "f-key template"   prg   2    "-----jm-----"      seq   7    "retirement.8"       prg   28   "copy-all v3.1"      prg
7    "dir to ptr.BAS"   prg   2    "-----ll-----"     seq   37   "inst.star.trek"    seq   9    "spheres.8"          prg   2    "+copy-a.21c3c"      prg
6    "status form.BAS"  prg   5    "universal wedge"  prg   11   "special.char"      seq   18   "payload.8"          prg   1    "-------------a+"    prg
1    "-- games --"      seq   23   "mail list 4.1"    prg   55   "nuke b128"         prg   18   "bar graphic.8"      prg   1    "--applications--"   prg
14   "caves of ice 128" prg   20   "mail list 2.0"    prg   10   "inst.B-Labyrinth"  seq   33   "rock trivia.8"      prg   1    "-------------a-"    prg
10   "states tutor.BAS" prg   12   "format"           prg   3    "-----dr-----"      seq   4    "b128/8050 speed"    prg   64   "autoexp v1.1"       prg
2    "-----dw-----"     seq   43   "update"           prg   12   "list-me (p)t7.1"   prg   1    "-----ww-----"       seq   70   "checkbook v1.2"     prg
7    "read first.ss"    seq   29   "print"            prg   4    "diskspeedtestb.z"  prg   22   "SKYLES MOD for B"   prg   1    "-------------w+"    prg
4    "chg ld addr.ss"   seq   3    "menu"             prg   3    "find type.z"       prg   5    "WOLF RS232 print"   seq   1    "--wordprocess--"    prg
11   "chg ld addr v1.w" prg   10   "mail backup"      seq   56   "super store.8"     prg   24   "SKYLES IEEE"        prg   1    "-------------w-"    prg
5    "G-Solitaire.ss"   seq   12   "journal"          prg   15   "flying charges.8"  prg   3    "WOLF letter.form"   seq   9    "typewriter v1.4"    prg
23   "G-Solitaire.w"    prg   23   "tax 80 ont v3.0"  prg   9    "unicopy inst.p"    prg   30   "skyles ieee"        seq   1    "   directory   "    seq
[illegible] "disk log b128.ss" seq 23 "mail list 4040"  prg   7    "unicopy4.p"        prg   4    "WOLF C64 to B128"   seq   3    "-----cz-----"       seq
16   "disk log b128.w"  prg   41   "4040 update"      prg   8    "savings accnt.8"   prg   22   "sys.0.cnfg.a3"      prg   17   "bterm.bas"          prg
7    "basic x-ref.ss"   seq   5    "4040 format"      prg   8    "loan payment.8"    prg   2    "-----jw-----"       seq   17   "bterm.pal"          prg
11   "basic x-ref.w"    prg   25   "4040 print"       prg   52   "the evaluator.p"   prg   17   ".menu"              prg   2    "bterm.obj"          prg
6    "directorysort.ss" seq   3    "4040 menu"        prg   11   "file create.p"     prg   1    ".directory"         seq   59 blocks free.
7    "directory sort.w" prg   23   "mail list 8050"   prg   37   "disp/avg/list.p"   prg   16   "timed typing II"    prg
4    "cassettelabel.ss" seq   6    "8050 format"      prg   48   "nomenclature.p"    prg   114  "project typing"     prg
```

TERMINAL #11, v1.0 CBUG #11.0 #12257

This disk contains the well known terminal program B-term, the most necessary of capture buffer programs for use in general telecommunications work. Of course, it is no match for B-line elsewhere mentioned. Also included is Teleterm 80 which was also distributed by Protecto, a reportedly public domain program including variations of Xmodem and the original Punter protocols (none of which seem to upload to major BBS services, but will download ok). In this disk you will find an xmodem program (xterm) in a 300 baud only version converted from the standard C64 original, and a sophisticated main-frame programmers' terminal program by Warren Swan (via the kind release from his royalty disk CBUG #3). There are several instruction files and articles, including those provided by the Toronto Pet Users Group concerning telecommunications. This is likely one of the most comprehensive telecom compendiums ever published. If you need help understanding the ins and outs of telecommunication all you need to get going like a pro is in the various articles. We'd have printed them, but it would have been a fair sized book!

```
1 "term w/bt 11 122" 25 2c   52   "READ MEMO"         seq   62   "Terminal.ins"      seq   18   "telecom canada"   seq   30   "cooper on modems" seq
[illegible] "loader"        prg   26   "terminal"          prg   3    "---tc info---"     seq   34   "online search"    seq   8    "rick's postscrip" seq
[illegible] "txram"         prg   5    "/term"             prg   12   "B-Function Key"    prg   18   "1book review"     seq   1    "---cbug misc---"  seq
[illegible] "keyscan"       prg   2    "teleterm80"        prg   2    "instr b-x modem"   seq   17   "how easy it is"   seq   19   "signon delphi"    seq
39   "term"                 prg   36   "o.teleterm80"      prg   3    "b funct key.ins"   seq   24   "on compuserve"    seq   38   "SS vs BT"         seq
3    "r"                    seq   1    " --INSTRUCTIONS-"  seq   6    "---TPUG---"        seq   34   "navigating cis"   seq   918  blocks free.
25   "xterm orig b128"      prg   89   "bterm.ins"         seq   42   "starting on delp"  seq   1    "---lc---"         seq
26   "cis xterm dl"         prg   58   "terminal.ins"      seq   27   "navigating delph"  seq   19   "cooper RS232"     seq
```

TERMINAL #11A, v1.Ø CBUG #11A.0 #12261

Same as above, but less royalty bearing B-Term and Teleterm 80

Scott's ML Monitor CBUG #12 #12984

Larry Scott has provided us with a reportedly far advanced professional machine language monitor. HOWEVER, this program requires the 24K RAM cartridge advertised elsewhere herein by Gary Anderson. The Ram Cartridge has many important uses which will be explained in future issues, so those of you into serious programming should not delay!

```
1 "bmon master     " bm 2c   77  "bmon inst.ss"      seq  20  "bmonp01.psrc"   prg  17  "bmonp06.psrc"    prg  17  "bmon.obj400 4000" prg
5   "bmon intro.bas"  prg    5   "bmon quickref.ss"  seq  17  "bmonp02.psrc"   prg  26  "bmonp07.psrc"    prg  1667 blocks free.
17  "bmon.obj3e00"    prg    20  "exp schem.bas"     prg  17  "bmonp03.psrc"   prg  20  "bmonp08.psrc"    prg
17  "bmon.obj4000"    prg    2   "bmonpct.psrc"      prg  15  "bmonp04.psrc"   prg  22  "bmonp09.psrc"    prg
17  "bmon.obj400"     prg    18  "bmonplb.psrc"      prg  19  "bmonp05.psrc"   prg  16  "bmonp10.psrc"    prg
```

PRINT FILES #3 CBUG #25 NEW RELEASE #12665

These are the print files from the CBUG January 1986 telecommunications issue, followed by the full text of Gary North's Electronic Mayhem Articles organized in both bisected and 4 part forms, the full text of William Wolf's Disk Care thesis, etc., Marilyn Gardiner's CBUG Local Newsletter and Anthony Liversidge's articles. Of course all the items subsequent to the January issue are full text of articles appearing herein.

```
1 "jan86 telecom  " j8 2c   33  "deb.bk"           seq  23  "order form 0186"   seq  2    "-----ww-----"     seq  41   "Dec85 nwsltr"       seq
20  "announc 0186 tel" seq  3   "scH"              seq  10  "jessie meeting"    seq  129  "WOLF disk tips"   seq  1    "-----al-----"       seq
83  "superspell.tm"   seq   88  "scratch 0186"     seq  1   "--- above ---"     seq  22   "SKYLES MOD for B" prg  5    "catalogue AL"       seq
7   "want ads 0186"   seq   1   "yell.mm"          seq  2   "-----gn-----"      seq  5    "WOLF RS232 print" seq  8    "star logo"          prg
9   "branca ltr"      seq   12  "yell.list"        seq  102 "1"                 seq  24   "SKYLES IEEE"      prg  8    "desiderata"         seq
6   "spurlock ltr"    seq   22  "yell.pt"          seq  106 "2"                 seq  3    "WOLF letter.form" seq  8    "keys"               prg
14  "letters.bk"      seq   35  "library 0186"     seq  64  "a"                 seq  30   "skyles ieee"      seq  1    "PS to reader"       seq
19  "letters.pt"      seq   7   "comp/trac ad"     seq  51  "b"                 seq  4    "WOLF C64 to B128" seq  10   "keys written"       seq
29  "be.online.pt"    seq   8   "comp/trac1 ad"    seq  49  "c"                 seq  22   "sys.0.cnfg.a3"    prg  119  "article"            seq
38  "deb.pt"          seq   8   "paid ads 0186"    seq  70  "d"                 seq  126  "disk care reshap" seq  112  "al-norm dec15 85"   seq
31  "deb.clean"       seq   2   "postage frank"    seq  3   "letter to reader"  seq  1    "-----mg-----"     seq  415 blocks free.
```

PRINT FILES #2 CBUG #26 NEW RELEASE #12651

These are the print files for this issue. Files are set up as "linked files" starting with "scratch 0286" and "library lead" so that you can do text search in Superscript for whatever information you may need -- thru the entire disk files in one shot!. Now that's even better than having an index!

```
1 "escape assy 2-86" es 2c  74  "rotello"          seq  37  "cooper"            seq  12  "13"               seq  4   "23"                 seq
61  "hints & tips"    seq   23  "uptain"           seq  35  "scott"             seq  20  "14"               seq  7   "24"                 seq
45  "lovejoy"         seq   59  "matos"            seq  14  "taylor"            seq  11  "15"               seq  25  "p4"                 seq
20  "ohalloran"       seq   60  "gardiner"         seq  39  "library lead"      seq  22  "7"                seq  15  "telecom lib"        seq
130 "liz"             seq   63  "wolfram"          seq  132 "scratch 0286"      seq  11  "16"               seq  9   "25"                 seq
29  "pawlus"          seq   47  "spurlock"         seq  13  "want ads"          seq  18  "3"                seq  10  "26"                 seq
25  "tingley"         seq   22  "wolf/liversidge"  seq  3   "wanted ads"        seq  16  "17"               seq  214 blocks free.
6   "sherman"         seq   85  "liversidge"       seq  9   "hohs"              seq  13  "18"               seq
53  "kernaghan dir"   seq   122 "swan"             seq  8   "harrison"          seq  21  "19"               seq
16  "kernaghan 8050"  seq   75  "kernaghan1"       seq  13  "table0contents"    seq  26  "20"               seq
16  "mills"           seq   44  "ames"             seq  121 "klinger"           seq  11  "21"               seq
36  "anderson"        seq   6   "wolf RS"          seq  22  "ads pg 60"         seq  24  "22"               seq
```

KERNAGHAN'S UTILITY CBUG #4 #12896

Warren did such a bang up job with this disk, that it was a best seller last fall. Here it is again, both as an example of how to do it right, and for the rest of you who missed out last time.

'Change 8050' changes the device number from 8 to 9 through 15.
'Chg addr 1' modifies the load address of programs.
'Compactor' will compact a program for faster running and less space.
'Compactor.fast' does the same, faster. Uses 8432 Pet emulator.
'Conv c64-b128' & 'conv petcbm-b128' are charts of memory addresses.
'Copy' copies files from drive 0 to drive 1.
'Copy 8/9 chg add' copies from device 8 to 9, while changing address.
'Copy all mult' formats a disk and copies all programs.
'Copy alpha v2' can format, copy all or selected files, and alphabetize.
'Copy-sort' copies all files from 0 to 1 with optional alpha sorting.
'Data writer' reads code in memory, then outputs it for data statements.
'Dir 2 col ptr' prints a 2-column directory.
'Dir printout' is an elegant little directory printer.
'Disassem pet sym' disassembles from memory, to screen or printer.
'Disassembler p' is a Butterfield disassembler for screen/printer.
'Disassembler.d' works with the files on disk, for screen/printer.
'Disinterpreter' converts code in memory to BASIC on screen.
'Disk checker' tests disks in device 8, drive 0.
'Disk commands' allows loading, scratching, renaming, or copying.
'Disk doctor' displays a sector of a specified track, allowing changes.
'Disk logger' provides a deluxe directory, with T/S and loading addresses.
'Disk maintenance' offers backup, copy, directory, new, rename, scratch,.
'Disk manager' has formatting, single program copying, renaming, erasing.
'Disk revival' is an unscratcher, to restore scratched files.
'Disk T/S' displays any track/sector on screen/printer.
'Disk tidier' helps delete unwanted files, using the directory.
"Disk unscratch' is an excellent file restorer for scratched files.
'Diskview' not only views, it can change what it shows you.
'Dsk header chg' will change the name of a disk.
'Dsk id nbr chg' will change the id characters of a disk.
'Hex dump dos' is a hex dump of disk drive processor or IEEE processor.
'Hex dumper 80' is a hex dumper from disk.
'Load address is' displays the load address of any program file.
'MSD purge.b' selectively scratches program files on an MSD drive.
'Peek print' displays the results of a keypress at any address location.
'pgm data.disk & pgm search.disk locate programs in your disk library.
'prog print' allows easy printing of programs.
'Prog.printout' prints program listings.
'Seq list cbm/ascii' reads seq files such as those on this disk.
'Seq lister' quickly displays a sequential file to screen.
'Seq prg-bas prg' tokenizes listings of basic programs.
'Seq read/print' is a sequential file reader and printer.
'Typewriter' uses computer keyboard and printer as a typewriter.
'Uncompactor' stretches those compacted or regular programs.
'WP prg printout' is a good program for printing most wordpro files.
'WP read/print' is good for displaying wp files, but prints slowly.
CALENDARS
'Calendar printer' outputs any month to screen or printer.
'Calendars screen' displays any month from 1904-2099.
'Day of week' outputs the day of the week from day, month, and year.
'Memory calendar' is an appointment record. (Uses 'March').
'Per calendar 2' will display any year and/or any month's calendar.
'Perp calendar' will display a calendar accurate from 1582 on.
EDUCATIONAL
'Body weight' calculates more than you can believe about your body.
'Brain balance' helps determine which side of your brain is most used.
'Life expectancy' will estimate your total span of years.
'Long division' drills students in progressively difficult problems.
'Yellow light' is a reaction time evaluation.
ENTERTAINMENT
'Bioprinter' furnishes screen or hardcopy for bioryhthm plots.
'Bioryhthm 2.5' displays the plots on screen only.
GAMES
'Afo/balloon' is a graphics game wherein the player tries to hit the AFO.
'Blackjack' is the old standby, with graphics.
'Blackjack-jb' takes place at the Butterfield Social & Recreational Club.
'Bomber' is a graphics game, with the player trying to down the bomber.
'Checkers80' is expanded to wide screen B128.
'Dominoes.c' is an easy to play graphics game.
'Hamurabi' Try your hand at governing Sumeria for a 10 year term.
'Kingdom' is an improved version similar to hamurabi.
'Labyrinth' lets you specify a 3-D labyrinth, then try to get through it.
'Lunar lander' is learning to balance velocity, fuel and time.
'Super lander' is an improved version of lunar lander.
GRAPHICS
'75 names' can be a grid filled with one character, or 80 columns repeating a name, phrase, or sentence.
'ASCII screen' pokes the screen RAM with any ASCII code typed in.
'Banner' outputs large characters or signs on the printer.
'Bar chart' is a demonstration of graphics display.
'Bar graph subrtn' is a demo intended to be interfaced with a program.
'Bar grf' is a horizontal bar graph presentation to screen/printer.
'Bargraph' is a comprehensive vertical bar auto-scaling printer.
'Big letter ads' displays, on screen, moving messages of up to 20 lines.
'Big letters' displays a brief message in 3-D characters.
'Big time' displays a large running digital clock.
'Pedigree charts' prints blank pedigree charts for geneologists.
'Squiggle' displays entertaining designs.
SOUND
'Yesterday' from the same song. Just 'run'.
TECHNICAL
'555 timer' helps design a timer circuit, for electronics design.
'Anglo-metric' is a conversion program, easy to use.
'Lo-pass filter' helps design a low-pass active filter circuit.
'Metric conv' is similar to 'anglo-metric'. Take your pick.
'Reg ps design' helps design a complete DC power supply.
'TV satellite' calculates satellite positions.
CHALLENGE
'Prod soft 2.7' is a disk drive diagnostic program that I'd appreciate help with in getting operational.
232 blocks free

LETTERS TO THE EDITOR

We can only afford space to print a very very few letters received. Two that I think ought to be snared are below. Most certainly, CBUG will honor its journalistic responsibility and give equal time to disenting opinions as well.

Excerpted from: Member Raymond B. Smith, Dunedin, Fl.

re: Enumerated OOPS

Opps #1: My system is down. (Rats!)

Opps #2: You want more money for "Knight" after you you sold it than you did before. (SIGH)

Opps #3 You sent me, in the Bakers 1/2 Dozen, a disk mis-labeled "Kernaghan's". Please send me a "Kernaghan's" & mis-label it whatever you wish -- (Sigh).

GOSH, Guys, CBUG is alabor of Love; I want you all to be able to OOPS for years - and to enjoy it. Enclosed is something for the crew some night when you're tired &/or discouraged.

Keep up the Semi-Good work!

- -

** CBUG MEMBER PERSONAL ADS **

Electronic Consulting Services: If you require professional assistance in the fields of: COMPUTER SYSTEMS-- for the personal and small business user; CONSUMER ELECTRONICS-- AUDIO- Stereo/HI-Fi for home and small professional installations // VIDEO-Projection TV/Monitors/VCR's for home /business // COMMUNICATIONS- ShortWave Listening/HAM Radio/CB Emergency Radio/49Mhz Talkies // TELEPHONES- Carriers: AT&T? GTE? MCI? ALLNET?/Buy? Rent?: Phones, Dialers, Answering Machines. For expert assistance at reasonable rates CALL--- Angel Matos at 212-231-6028.

* * * * * * * *

The Home Office Newsletter

Member Joseph Rotello is entering the publishing business with an entry of interest to many CBUGer's. He will be publishing a quarterly research paper on the subject titled above. Write Joe for information at: 4734 E. 26th St., Tucson, Az. 85711.

January 12, 1986

Mr. Norman Deltskee
c/o CBUGS in Chicago
4102 North Odell
Norridge, Illinois 60634 **Attention: Irate Customer Department**

Dear Mr. Deltskee:

You really have spoiled my New Year. I am writing you to tell you people that I have just had enough. I was enjoying my retirement here in Sun City ... playing an occasional game on my Commodore 64 ... until I sent away for that Protecto offer ... the "new" B-128 computer package, and sold my C-64. Things were bad enough when I somehow destroyed my Info Designs inventory disk without really trying. BUT ... I really got mad when I sent and received that CBUG Financial Utilities program diskette of yours. It is not fair to put out such a humungous, horrendous disk. All those programs on one diskette. Hell! I have trouble enough getting one program on a disk to work at one time. The whole disk is full of all kinds of things. It just boggles the mind. Frankly, it is just too much for me. I am selling my B-128 system and going to buy my old C-64 back. Then when I get nervous ... I can sit back and play a few games and relax. Your dam disk is responsible for my troubles. No one should put that many programs and that much information on one disk. There out to be a law against it.

Sincerely,

Dimmie

Shacho V. "Dimmie" Benidetto
69 Delphi Lane
Sun City, Arizona

P.S. I have cutup and enclosed this disk I bought from you. Please cancel my account and membership in your club. I don°t want to be a Bugger anymore. If you°re lucky you won°t hear from me again ... but you never know.

ABOUT 6400 PRINTERS w/ CENTRONICS PORTING

Reconvert back to IEEE !!!

There is availability of the excellent CBM 6400 printer at near give-away prices as mentioned in the want ad section. However these printers, as well as many our members now have are equiped with Centronics porting instead of IEEE. This requires the use of a converter -- and some inconvenience and expense. CBUG, with the very kind cooperation of CBM, has been able to acquire a quantity of the original IEEE port cards and internal cables. From all appearances, the interchange is but 5 minutes work with a screw driver only. Just take the cover off, swap the boards and a single screw holds the plug assembly in place. Call us about the great opportunity after April 21 1986 -- by which time we will should have the boards in stock and some instructions written. They should be substantially less than $40.00 each.
Norman Deltzke, CBUG, 312 456 8720, Tue. thru Sat. 7pm to 10pm CST.

it. Be sure that you have a backup copy of the disk, if you mess up while using 'disk mod' there's no way back !!!

Put the disk that you want to modify back in the drive and enter the track and sector in decimal numbers of the starting block of the program.

You should see something that looks like this sample:

```
00 : 13 10 01 08 0f 04 64 00 54 43 b2 54 49 aa 32 30 :        $ tc=ti+20
```

Let's go through this line piece at a time.

1. 00 : - this is the location of this information within the track and sector selected. Don't worry about it.
2. 13 10 - this is hex address of the next track and sector of the program. You don't need this information.
3. 01 08 - this is the address where the computer will load this basic program !!!!! Our goal at last !!!!! This hex number is supplied in reverse order ie. the hi byte is first. In this example, this number translates to $0801 or $801. This is the starting address of basic in all C-64 computers.
4. 0f 04 - information the computer uses, don't worry about it.
5. 64 00 - basic line number of first program line. You don't need this information.
6. 54 43 b2 54 49 aa 32 30 : - these are the hex ascii codes for the letters printed at the end of the line. You don't need this information.

Now we're ready to change the address. Use your cursor controls to move the flashing cursor over the number we need to change, in this case the fourth byte of the first line '08'. Type over the numbers that are there now to make this byte read '04' to make Liz Deal happy or '03' if you want the actual basic starting address of the B-128 computer. Either value is okay, but using Liz's address will prevent problems with DOS wedges, machine code transfer routines etc. in the future. Remembering that 'Hell hath no fury like a woman scorned', I would recommend the '04' starting address also. It will also insure that if someone ever uses your program that is using the utilities packages that they won't have any problems with it.

Now use the cusor controls to move the cursor to the end of the line, in the blank space after the 16th byte and the ':'. Now hit <return>. The disk drive should startup and the screen go blank. After the drive stops, the screen should rewrite itself and display the first line with '04' in the fourth byte position. Congratulations, you just changed the program loading address !!!

If you have any more programs on this disk that you want to work on, you should move on to the next track and sector location and repeat the above for each.

THE THIRD STEP

In the unlikely even that the program you transferred had no peeks or pokes in it, you're all done. Go ahead and load and run it. It should work perfectly. However, it will display on the left hand side of the screen only, assuming that it came from a C-64 or early 2000, 4000 or 8000 machine. This is because all of these machines are only 40 column display. To fix this problem, if you want to, add the following lines of basic to the beginning of the program or write a 'boot' program for your disk that contains these lines.

```
1 print"[home][home][clear-home][normal]"
2 print"[clear-home]"tab(20)chr$(15)
3 print"[24 X down cursor]"tab(60)chr$(143)
```

This creates a 40 column window in the center of the screen which centers the output of a program from an 8032 or C-64 computer.

If your program did contain some peeks and pokes, it is now time to fix them. To fix the peeks and pokes you will need a copy of the C-64 'Programmers Reference Guide', Jim Butterfield's 'Machine Language for the Commodore 64 and other Commodore Computers' and Protecto's 'Commodore B-128 Programmers Reference Guide'. The process of repair is relatively simple. Look up the poke or peek code meaning in the manual for the source computer (the computer that the program was designed to run on originally) and then find the equivalent code in the manual for the B-128. Substitute the proper code in the program and then save the result.

HELPFUL HINTS

Since the B-128 doesn't support any sort of color graphics, any calls, peeks or pokes into the color areas of the C-64 can be erased with no ill effects. Be careful to not eliminate any logic that may have been used by poking or peeking at color screen locations. If logic was used, then relocate the pokes and peeks to some unused B-128 location like around 800 decimal on bank 15.

Clock routines must always be modified. On the B-128, 'ti' is not a reserved variable for the jiffy timer. Therefore you must add a line somewhere that defines 'ti' as equal to val(ti$). Also the value of 'ti$' contains 7 digits rather than the 6 used in the C-64 or 8032. What this means is that any value counted or divided into 'ti' should be multiplied by 10 to restore proper clock function.

Graphics, upper and lower case letters. In most cases you will need to add print statements using the 'normal' or 'graphics' symbol as displayed in quotes mode to get the proper screen display from the program.

CONCLUSION

I hope that this article has been useful to you in getting started in the conversion of C-64 and 8032 software for operation on the B-128 computer. If you run into any unusual problems, drop me a line at SCCBMUG at 19 Fairway, Cocoa Beach, FL 32931.

In the next article I will attempt to discuss the conversion of simple machine code programs using the Commodore machine code transfer routine and software from the C-64 and 8032.

Bibliography

Programs - Jim Butterfield "Disk Mod" Cal. BUG utility disk # 2 and published in text form in Cal BUG issue # 4
- Jim Butterfield "Copy.all.rev" available from CBUG
- TPUG "8432 emulator" available from CBUG
- TPUG "scopy5" available from CBUG (must be used with emulator)
- TPUG "change unit addr" available from CBUG

- TPUG "pgm cross ref" available from CBUG
- TPUG "X-modem" available from CBUG

NOTE - All of the above are also available from the SCCBMUG user group on a single disk for $ 10

Publications - Commodore 64 Programmer's Reference Guide (about $ 12 at K-Mart, Walmart, etc.)
- Commodore B-128 Programmers Reference Guide (about $ 20 from Protecto)
- Machine Language for the Commodore 64 and other Commodore Computers (about $ 15 at Walden Books or B Dalton) written by Jim Butterfield

GARY NORTH'S REMNANT REVIEW-ELCTRONIC AIDS

Edited by Barbara Spurlock

"what You are about to read will shock you....as it did me as I did research on the project." So writes Gary North in GARY NORTH'S REMNANT REVIEW, vol. 12, No. 20, November I, I985, with a follow up Nov. I5, which he aptly entitled: ELECTRONIC AIDS (PART 2). What North so convincingly writes about are the threats the Western nations face as they become increasingly dependent on computerized technology. As the technology of computers becomes more sophisticated so do the users. What happens to the security of nations if the youths who now tamper so freely with computer systems in the spirit of challenge, fun and game, become the electronic thieves, terrorists or blackmailers of tomorrow. Open to potential attack are all systems that use common carriers: banks, national defense, large and small businesses, public utilities and the voting polls.

To illustrate his point, North gives us an interesting scenario which basically is this: By telegram, unknown terrorists inform the committe of the Federal Open Market, which sets U.S. monetary policy, that, of this day, four rural banks will find all their data scrambled beyond redemption. This happens.

The press has been tipped off. The governors of the affected states, fearing a rush on all their state's banks, demand action.

Then the FOM committee, again by telegram, is informed that, unless the U.S. breaks off dipomatic relations with Israel, three (named) major banks will loose all their data. The telegram continues: "TO PROVE WE CAN DO THIS WE WILL SCRAMBLE THE RECORDS AT CHASE MANHATTAN BRANCH BANK XYZ TOMORROW MORNING STOP."

The next morning, all of the records of Chase Manhattan's branch bank are turned into random numbers. That afternoon, the President of United States breaks off diplomatic relations with Israel.

Science Fiction? Reality? It's fiction that lies in the realm of reality.

So concerned is North of such grave implications and possible news cover-ups, he is going public with his information, lifting the copyrights to his articles, and offering $1,000 to anyone who can prove, in a taped interview, his information incorrect. He "desperately wants to be proven wrong."

North, an Economist, literally stumbled on the information he has. Reading how "hackers" were accessing computer systems and causing havoc, how computer software companies, in order to protect their programs from piracy, implanted "worms" into their programs, how young computer freaks were playing a game called Core War that broke down the security defenses of their opponents programs, North concluded the same technics used by hackers, could be used for things much more dangerous than games.

Greatly concerned, North, not a computer expert himself, consulted a friend who is. North fictitiously calls his friend Tom in order to protect his friends identity and his job. North describes Tom as a "space cadet"-a genius in the world of computers. Previously employed to set up programs used in the U.S. banking system, Tom now heads a firm which supplies computer services. In a 90 minute taped interview with North, Tom details security risks to computer systems.

Worms: Worms are undetectable commands that are implanted in programs and are activated when the programs are copied. What do worms do? They eat things. They can eat programs that have taken year to put together. They can spread throughout the users whole computer system.

Jackpotting: Banks are in a rush to install computer controlled automated teller machines where ever they can to get a corner of the market. These ATM's are hooked up to banks by a telephone line. By accessing the line by computer, an electronic thief can get the ATM to empty itself. ATM thieves ARE hitting banks continually, but nothing shows up in the press. "At present, there is no known defense." says Tom.

Rounding off: A bank's computer can handle only thirteen digits. Numbers are routinely rounded off. In major transactions involving millions, the rounded-off figure can be considerable. Insiders, with access to the computer programs, have stashed away hundreds of thousands of dollars--maybe millions--over a few years. And yet that siphoning off of funds can go undetected. The bank's books balance-to their rounded off figure.

Arsonists: These are people that access programs through telephone connected-computers. How does it work? An arsonist hooks his computer to a phone line (loop) and randomly dials numbers until the tell-tale whine of a computer line is heard and the number recorded. By creating a deliberate error, the protective shell of the program brakes down and the arsonist has access to the whole program. This can be done from all over the world, free of charge, with a device called a black box. Programs that enable phone access are sold. Some people get caught, but it's the "tip of the iceberg," adds Tom."

What can the arsonists do? They can steal money, or threaten to steal money as a form of blackmail. They can create a new identity for themselves or change their school grades. They also can plant viruses.

Viruses: viruses are small, untraceable commands that arsonist, terrorist or blackmailers can implant into a computer system. In a bank, they can spread to any other bank's computer by means of electronic funds transfer. Once inside, viruses can infest the whole system, devour or scramble all data, or any other task it's programed to do. They can be activated, say, when someone calls into a bank system with an uncommon name or a key word. A virus is a computer-to-computer killer, where as a worm is a resident in one man's computer system.

Can it happen here? Tom says it would be a piece of cake for him to rob a bank of three or four million and then demand 10% of it back as a finders fee for the return of the money. They would probably pay just to keep me quiet. Adam Osborne, the inventor of the revolutionary portable computer, the Osborn I, and the author of RUNNING WILD, a history book on microcomputers, states: "There are three areas that should never be computerized: bank money transfers; the stock market; elections." Nearly all are now computerized. "The great fortunes of the 2Ist century," Osborne predicts, "will be the legacies of the great computer thieves of the 20th."

In July, I985 twenty three teen-aged computer users broke into a Chase Manhattan Bank computer and caused "significant damage" to bank records by changing bank passwords and preventing customers-and in one case, a unit of the bank itself-access to their own files. Interactive Data Corp. of Waltham, Massachusetts, which maintains the bank's financial records, has over 25,000 subscribers to their services. Now, Chase Manhattan bank, Citicorp, and Bank of America are adding expanded services so their customers can pay monthly bills electronically, balance their "checkbook", monitor their net worth, buy and sell stocks. With expanded services comes expanded vulnerability.

A reporter for WASHINGTON TIMES bought a newly released program from the Microsoft Company. Microsoft supplies the disk operating system which is used by the IBM PC, the most popular microcomputer. The first time the reporter tried to use it, a warning flashed on his screen: "The weed of crime bears bitter fruit. Now trashing your program disk." Wham! He lost all his files-probably a couple of year's worth of work. Microsoft admitted an unauthorized programmer had inserted the worm.

A high school student in Pittsburgh wrote a virus that infected not only his disks, but those of his friends and his math teacher's graphing disk. He wrote a program to destroy the virus, (such a program is called antidote or antibody, a counter-virus agent which seeks out and destroys a specific virus. They can only be created by those who know what kind of virus is involved), but the virus was too strong to be destroyed by the antidote.

About four years ago, North happened to read an article in Rolling Stone magazine which described a subculture of student "hackers" at Stanford University. These students, some of whom were libertarians-anarchist-used the University's mainframe computer from midnight until 6 a.m. when the use charges were cheapest. Besides working on their own programs, they also played a game called Core War. The purpose of this game was to build up an impermeable defense system for one's own programs, AND, to try and knock out the defense systems of other users. It was considered the mark of a master hacker to be able to crack another hacker's defenses. They were "hacker-crackers."

One enterprising young man designed a booby-trap. The program warned would-be raiders not to tamper. That warning only served to make cracking the program more of a challenge. For days, other players tried. Finally, one succeeded. Then on his own screen appeared the message: "Now you've had it," and he watched in horror as the names of his own programs that had taken years to assemble appeared on screen and were erased, one by one.

Actually, it had been a hoax. The victim hadn't lost all his files, only the names, but, by repeating this story for us, North made some important points: Young computer users like to play games. The better the challenge, the more the fun. Computer use isn't confined necessarily to honorable people. The games they play can be dangerous.

North goes on to say; "These people are geniuses....in a very narrow field, technically, which is now being used to control darned near everything. This unique intellectual-technical skill is the possession of literally a handful of people, mostly under 35 years of age. They are 'fooling around' with Chase Manhattan Bank's computers. What happens when a few of them stop fooling around and get deadly serious?"

North concludes his articles with the following: Technologically, there is no solution at this point. I have no heartening message. Maybe later; not now. Keep precious metal coins. Don't assume that it can't happen here. It can. The only thing holding it back is the restraining hand of God, through the temporary self-restraint of a technological priesthood.

<<We have included this abridged article as many have found it of interest, even though it has nothing to do with the B128. Barbara has kindly reduced a huge file so you could get a peek. The entire text is on the Print File #2 disk in Library under the name "mayhem". The entire file is also available in the Data Library in Flagship Commodore within the Delphi network. We also thank member Welbourne Tooker who took the time to transcribe the two articles of about 30 pages in all to 8050 disk for us.>>

NOTE: The opinions and information stated in this article are those of Mr. Gary North and he alone. No independent verification has been undertaken in the course of presenting the above to CBUG members.

- - - - - - - - -

about Mrs. Spurlock

Barbara Spurlock is a native of Wyoming and now resides in Lander, which she calls "God's Country." She attended the University of Colorado-Boulder-and has continued her education at a near-by community college. After her five children were grown, Barbara decided to start a career in writing--taking after her mother, Shannon Garst, author of forty-four books. She gave up her position of over fourteen years on the board of Trustees at Central Wyoming College to take on a new role with the college, student. Barbara now has a column and writes articles for the WYOMING STATE JOURNAL. She is also a member of the B128 Users Group.

'THAT AIN'T NO FRISBEE'

A Thesis On Disk Care

from text by William H. Wolf
abridged by Anthony Liversidge

Diskettes are suprisingly resiliant, and they hold up well in daily use. So it's easy to get careless. But it only takes one accident to bring home that even small erasures can be catastrophic. CBUG member William Wolf has written an in-depth 4,800 word analysis on How to protect your Data Investments. It's a complete rundown on all threats to your disks' health and welfare, including many you wouldn't otherwise think of until too late. We can present only a summary in this newsletter. For the full explication, the entire file is on the print file disk under "disk care" and will be in the CBUG's Encyclopedia Disk. <<The full text is well worth reading.>>

Here are Bill's main points:

Floppy disks are made up of jacket, internal wiper system, ferric oxide (rust, as in your car), integrated lubricants, binder glue and polyester disk. This makeup means they are vulnerable in four ways:

Mechanical: Warped or bent disks can spoil the timing of the read-write head by exerting drag and preventing the intimate contact needed to read the data. Writing on labels on a disk with ballpoint, using paper clips, even big plastic ones, or just bending the disk when inserting it into the drive can dimple or crease the disk surface, destroying data or access to data. Heavy objects - scissors and repair tools on or near your desk - dropped can knock out the disk surface by dimpling or, placed absentmindedly on top of a disk they can crush the jacket edges, causing wear, drag and data loss. Don't use rubber bands around disks., and don't even pile disks flat on top of each other! Store disks in vertical position.

Environment : Extremes of heat and cold - under 20 degrees and over 125 degrees F. - spell danger! Don't bring in a disk from icy weather and use it immediately - you'll lose data, because its rigidity will stress the drive heads. Warm the disk and dry it out to guard against condensation. Ideal temperature is 65 degrees F. and 40 per cent relative humidity. Avoid direct sunlight! Temperature over 125 F. - easily reached in a closed automobile in hot weather! - will distort the jacket and disk. High humidity makes disk coatings gummy, stalling and destroying your drives! Disk storage limits are 40-90 F. range, 20-80 % humidity.

Magnetic and radio fields: Magnetised items can forever destroy data on disks instantly! Keep refrigerator magnets, audio equipment, loud speakers, pocket radios, fluorescent desk lamps, unshielded electric motors at least three inches AWAY! Paper clips are usually harmless, though not always. If data suffers only weakened signal amplitude, copying to another disk can restore it. Electromagnetic radiation can affect disks only if very strong i.e. if disk is very close to antenna, cables, or transmitter of radio or radar (no loss observed in disks placed ten feet away from 500,000 watt transmitter). Radio waves can produce strange bugs in computer and disk drive, though, causing inexplicable read/write arrors.

Physical conamination: Do not touch exposed disk surfaces - oil from fingers! Cutting notches in disks to use both sides is a foolish economy - it generates debris and anyway turning over a singlesided disk reverses the rotation past the internal sweeping system, releasing debris which will screw up the head and the disk. Don't even insert disks upside down by mistake for same reason. Preferably write on labels before affixing them, or use felt tip pens. Don't place labels over labels, or allow loose labels - may jam drive. Don't place tape over any holes in disk. Don't use eraser on disk - generates debris. No smoking or eating near computer for same reason - debris: even minute particles can destroy head contact. Heavy smokers should use air filtration system. Dust like a demon around computer - one speck of dust only 500 micromm in diameter can cause a dropout.

Acclimatise disks before archival use. Use zip lock bags. Maintain dead air space around files. Keep a fire extinguisher handy to your equipment.

Rescuing contaminated disks: Bill provides a special step-by-step briefing on how to rescue vital data from contaminated disks, using isopropyl alcohol, lint free gloves, liquid detergent or other cleaner, and a pair of panty hose to stretch over a bowl for a surface on which to dry the disk.

Congratulations are due Bill for a thorough job - he did exhaustive independent research, including personal interviews with industry experts, and reading research reports from the Federal Government, diskette manufacturers and independent engineers. The thesis is worth reading in full.

Abridged for the CBUG ESCAPE by Anthony Liversidge, NYC.
(c) 1986

<<The original of this article is 129 blocks long on disk. It is copyrighted by William H. Wolf, 1690 Hosmer Rd., Appleton, NY 14008. Mr. Wolf has kindly grated CBUG the privledge of providing this information to our membership ONLY. Any other reproduction in full or in part including the abridgement above is specifically prohibited.>>

SUPERSCRIPT IN HYPERDRIVE

by: Anthony Liversidge

At $100 (let alone $10, Protecto's final price) SuperScript on the B 128 is a bargain. A professional level program, SScript will carry out nearly all key writing, editing and production tasks faster and more efficiently than rivals costing hundreds of dollars more, including the most widely used program, WordStar.

The power of even the most expensive text processing software for personal computers is almost universally compromised by awkward design and garbled command structures. SuperSscript's designer, Simon Tranmer, is said to have aimed at outdoing Word Pro Four Plus, a snappy yet muscular program sold for CBM 8032s, the previous Commodore business machine. The result is a streamlined, powerful program easy to learn and to run at the expert level. The commands are mnemonic and logical, and the speed of execution is high (though unnecessarily compromised in one action, transferring a block within a large text).

Comparison with other programs will warm the heart of any Superscript user, even at this late stage in the history of word processing software. For example, the November 1985 Byte reviews the Sanyo MBC 775, a $2599 portable, and reveals a not uncommon limitation - EasyWrite II users can't find out what is on their disk without exiting the program! WordPerfect 4.1 at $495 retail is currently rated as "best at any price" by InfoWorld, and indeed it does outgun SuperScript in some features (triple undelete function, split screen, large thesaurus, search all text files for word combinations, sort function, alphabetization, indexing, outlines, table of contents.) A complete list of how Superscript might be improved is detailed and is presented in the full text version print file disk.

But much of what has just been boastfully added in the WordPerfect update is standard with SuperScript: center and flush right on entire blocks of text, side by side column display, see on screen what you get on printout, quick copying of files to another drive, retrieve, delete, or rename without leaving the program, uncluttered screen, informs you if disk has write protect tab, or needs formatting. And while WordPerfect is said to be second only to Xywrite II Plus in speed, one wonders if it is any faster than Superscript. Reviewers who think that Wordstar's two minutes to change margin on lengthy text in an IBM PC is the standard might have to be told that Superscript can do it in one second. (Anyone out there have hands on experience with Xywrite or WordPerfect 4.1?)

Meanwhile as IBM DisplayWriter still trumpets pathetically of 'automatic pagination' in its latest Wall Street Journal ads. One Superscript feature - range column block move, the power to move specified areas of the screen bodily in ny direction - is as far as we know otherwise confined to dedicated word processors.

DEFINING NEW FUNCTION KEYS

For those familiar with SuperScript, there's a way to make it much more effective: redefine the function keys. Most are programmed by SuperScript to save only one key stroke, and can be more usefully defined.

For example, Esc + F is on Function Key 13, Esc + L on Function Key 8, and so on. Obviously, this doesn't effectively exploit the function key potential. What is needed is Function Keys which incorporate long strings of keystrokes, not two strokes more easily made on the usual keys.

Proper loading of SuperScript should, in the US, begin with loading Elizabeth Deal's cursor-fix program "pre-superscript" on CBUG 'Liz Deal' disk to cure the cursor hesitation which otherwise plagues the B 128 used on power lines in the USA (it is due to the sixty cycles per second AC current used here, against the fifty cycles per second of UK power). So obtain a copy of "pre-superscript" from CBUG and load from Drive 0 as follows:

```
                    dload"pre-superscript", d0.
```

Now list the program (type list and press RETURN) and amend lines 1110 to 1120 as follows, by typing in alterations and pressing RETURN with the cursor beside the newly altered line:

```
 1110 print"qrNow use instruction below to load the hyperdrive magic of YOUR NAME revised keys - put cursor on the :::::: line and press RETURN"
 1120 q$=chr$(34):sysa:print"qq::::dload"q$"keys"q$",d1"
```

Save this revised cursor 'fix' program to a disk in drive 1, under the name 'fix'. (Type dsave"fix",d1 and press RETURN).

When you load SScript, you will first load and run this revised "fix" program, which will cure the cursor and then call up your revised 'keys' program, which is your own adaptation of the original function keys specifications from the SScript disk.

Revising function keys

How to prepare revised function keys? Load, from the SScript disk in Drive 0, the 'function keys' program lines 1000-1200, as per instructions on page F-1 of the manual. (In Basic, with SScript disc in Drive 0, type dload"superscript II" and Return, then type list 1000-1200 and Return.) Change their specifications to suit your own needs, and place them as a new program, 'keys', on the disk in Drive 1. When loading SScript, the 'keys' program will be loaded after the 'fix' program, the revised "pre-superscript", and before the main SScript program.

Amend the function keys listed from 1010-1200 as follows. These suggestions are my current best sustitutions having used SuperScript for two million words. Your tastes will differ, however, depending on level of expertise, type of work being carried out, etc.

The important point is that function keys as programmed by SScript mostly save only one keystroke, and are often more easily carried out as normal command sequences. Hitting Esc + F from the usual hands position is quicker and easier than hitting shifted F3, for example.

Ideally, a function keystroke will substitute for the maximum number of other keystrokes. Each substitution noted here saves more than one keystroke, and some as many as nine. Each replaces a too simple SScript function key definition which can be better carried out by a normal two stroke command, Esc+x.

(Note: One minor complication: when typing in new material in Basic, reverse highlight letters and figures are produced by the cursor keys if you enter quote mode by writing ", and also if you use Shift-delete to close up text (you enter Insert mode for as many characters as you close up). Cancel this by tapping the Esc key once. The Insert mode (produced by Shift-Ins) also produces highlighted characters on the cursor keys (Q,q,, and] highlighted) but only for as many characters as the key is tapped), so you don't need to cancel it.)

```
-------F1: 1010 key1,chr$(27)+"ov"   :rem  f1 = output to video
```

Unchanged. Saves two keystrokes, and is essential function.

```
-------F2: 1020 key2,chr$(27)+"hm"   :rem  f2 = hunt in memory
```

Moved here from F7 for layout reasons. This quick hunt function can be used to move rapidy from top of file to top of file in the same text area, if each file is headed with *nb"File Name", and the Search function is used to search for occurrences of *nb".

```
-------F3: 1030 key3,chr$(146)+"$0:" :rem  f3 = choose file from 0
```

Saves four keystrokes: Handy for rapid searching for specific file or files in disc mode on Drive 0, which contains your working disk. If Return pressed after it, it wil give you a quick reading on which disk is in Drive 0.

```
-------F4:  1040 key4,chr$(18)+"nb"   :rem  f4 = *nb comment line
```

Saves three keystrokes, inserting *nb whenever needed. Most useful when an entire paragraph must be hidden from printout, this can be simply added to the start of each line by tapping F4 once.

```
-------F5: 1050 key5,chr$(146)+"$0"+chr$(13)   :rem  f5 = view directory 0
```

This puts you straight into directory for disk in drive 0, running off the list of files at a tap of F5 (halt the unreeling with the chicken key, C=). Saves four keystrokes.

```
-------F6:  1060 key6,chr$(27)+"th"   :rem  f6 = set horizontal tab
```

One tap of F6 sets a horizontal tab at the cursor position, saving two strokes.

```
-------F7:  1070 key7,chr$(18)+"ps"+chr$(58)+"pause"+chr$(13):rem  f7 =*ps: pause para
```

At a tap of F7, this inserts *ps:pause and Return at the beginning of a line. This interrupts printing, so that you can print out a file at the top of a text area, for example, without worrying that the printer will go onto the next area of text in memory.

```
-------F8: 1080 key8,chr$(18)+"ma7"                    :rem  f8 = print *ma7
```

```
-------F8: 1080 key8,chr$(18)+"ma7"                    :rem  f8 = print *ma7
```

This allows you to conveniently indent text (after first changing *lm command to the value sought for the indented margin) with one stroke additions of this command at the end of each line you want to project to the full left margin of the text. Saves innumerable keystrokes per operation - invaluable for speeding up an otherwise laborious process.

```
-------F9:  1090 key9,chr$(27)+"u"+chr$(23)+chr$(27)+"u"   :rem  f9 = change case of word.
```

Carries out five keystrokes at once, changing the case of any word from the cursor position to end, and moving cursor to first letter of next word. Little increase in speed.

```
-------F10: 1100 key10,chr$(23)    :rem f10 = next word
```

Unchanged for same reason. It substitutes only for Cntrl W, which is equally convenient, but matches F20 (previous word).

```
-------F11: 1110 key11,chr$(27)+"O"  :rem f11 = re-output to video
```

Unchanged. A vital function to have readily available.

-------F12: 1120 key12,chr$(18)+"fp0"+ chr$(13) :rem f12 = *print fp0 para

A tap of F12 inserts *fp0 and a Return at the beginning of a line, handy for interrupting output of files, when more than one file is loaded into same text area. In video output, for example, if each file ends with a *fp0, each will be video displayed starting from the top of the page. If editing is carried out, even on a file deep into the text area, re-output displays it from the top again, almost as if it was in a separate memory area.

-------F13: 1130 key13,chr$(146)+"$1:"   :rem f13 = choose file from 1

Convenient immediate set up to ask for one file listing from directory 1. If Return pressed after this key, quick check of which disk in that drive.

-------F14: 1140 key14,chr$(27)+"G"+chr$(29)+chr$(29) :rem f14 = go to end of text, two spaces

This commands th cursor to go to the last letter at the end of the text, then move tuwo positions forward ready for additional text. (SScript untidily programs this function key to place cursor on last letter, necessitating another two keystrokes before writing can continue)

------ F15: 1150 key15,chr$(146)+"$1"+chr$(13) :rem f15 = view directory 1

Save three keystrokes to get immediately listing of Drive 1 directory.

------ F16: 1160 key16,chr$(27)+"ch" :rem f16 = clear horizontal tab

Quick cancellation of horizontal tab at cursor position, conveniently placed above set tab key F6.

-------F17: 1170 key13,chr$(27)+"zh"   :rem f17 = clear all horizontal tabs

Convenient clearing of all horizontal tabs.

-------F18:   1180   key18,chr$(18)+"lm1:rm80:ju0:sp0:ra0:cn0:vp0:pr0:pl66:tl53"+chr$(13)+chr$(18)+"hl1:hr80:hd0:,"+chr$(27)+chr$(35)+","+chr$(13)+chr$(18)+"hs3:,,"+chr$(13)

1185 rem-above is f 18 = complete standard heading

One tap of this key gives you a full standard heading. Set your own standard values on this one. Saves 71 (count'em) keystrokes, and doesn't wipe out your text as does the alternative of storing this in a file and inserting it, if you happen no to be in insert mode by somne error.

-------F19: 1190 key19,chr$(18)+"ft0:,,"+chr$(13)+chr$(18)+"fs9:,,"+chr$(13)   :rem f19 = footings

Same thing as above for footings. One keystroke sets up two footings.

-------F20: 1200 key20,chr$(157)+chr$(5)+chr$(29) :rem f20 = space after previous word

Places cursor on space after final letter of previous word, rather than on last letter as programmed inconveniently by SScript.

You may prefer other sequences on these keys. The choice of alternatives is limited by your imagination. Not many SScript commands have more than two keystrokes (Esc + x), so few commands deserve function key treatment. But with 160 letter spaces available for eac definition (that's how long a program line the B will read) other long printer command sequences may be candidates. Please inform this writer if you come up with improvements.

Once these revised keys are complete, save to disk on drive 1 as "keys", in company with "pre-superscript". (Type dsave"keys",d1 and press RETURN).

This is what the program should look like, if all the above suggestions are taken:

```
 100 printchr$(142):bload"star logo",d1,b15,p53248
 110 rem
 120 rem  your company logo may be substituted for the file ss2logo
 130 rem
 140 rem  the file is a screen dump in graphics mode...
 150 rem
 160 bank15:poke193,254:rem put function keys at top of memory
 170 rem
 1010 key1,chr$(27)+"ov"                     :rem  f1 = output to video
 1020 key2,chr$(27)+"hm"                     :rem  f2 = hunt in memory
 1030 key3,chr$(146)+"$0:"                   :rem  f3 = choose file from 0
 1040 key4,chr$(18)+"nb"                     :rem  f4 = *nb comment
 1050 key5,chr$(146)+"$0"+chr$(13)           :rem  f5 = view directory 0
 1060 key6,chr$(27)+"th"                     :rem  f6 = set horizontal tab
 1070 key7,chr$(18)+"ps"+chr$(58)+"pause"+chr$(13):rem  f7 = *ps:pause para
 1080 key8,chr$(18)+"ma7"                    :rem  f8 = print *ma7
 1090 key9,chr$(27)+"u"+chr$(23)+chr$(27)+"u"  :rem  f9 = change case of word
 1100 key10,chr$(23)                         :rem f10 = next word
 1105 rem
 1110 key11,chr$(27)+"O"                     :rem f11 = re-output to video
 1120 key12,chr$(18)+"fp0"+chr$(13)          :rem f12 = print *fp0 para
 1130 key13,chr$(146)+"$1:"                  :rem f13 = choose file from 1
 1140 key14,chr$(27)+"G"+chr$(29)+chr$(29) :rem f14 = go to end of text spaces
 1150 key15,chr$(146)+"$1"+chr$(13)          :rem f15 = view directory 1
 1160 key16,chr$(27)+"ch"                    :rem f16 = clear horizontal tab
 1170 key17,chr$(27)+"zh"                    :rem f17 = clear all horizontal tabs
 1180 key18,chr$(18)+"lm1:rm80:ju0:sp0:ra0:cn0:vp0:pr0:pl66:tl53"+chr$(13)+chr$
(18)+"hl1:hr80:hd0:,"+chr$(27)+chr$(35)+","+chr$(13)+chr$(18)+"hs3:,,"+chr$(13)
 1185 rem-above is f 18 = complete standard heading
 1190 key19,chr$(18)+"ft0:,,"+chr$(13)+chr$(18)+"fs9:,,"+chr$(13)
:rem f19 = footings
 1200 key20,chr$(157)+chr$(5)+chr$(29)   :rem f20 = to space after previous word
 1350 rem
 1360 rem function keys now all defined
 1370 rem
 1380 poke2045,255:rem setup cbm user-defined char 0 = no setup
 1390 poke2046,239:rem and with status for printer error checking
 1400 poke2047,0:rem make previous defaults effective
 1470 rem
 1480 rem  the following instruction will load superscript ii
 1490 rem
 1500 bload"load",b15,p256
ready.
```

Loading all three programs - 'fix', 'keys' and SScript - now is easy:

1. Type this: dload"fix",d1   and press RETURN.
2. Place cursor on the line beginning :::::: and press RETURN
3. Type this: run   and press RETURN.

that's all you have to do to boot hyperdrive text processing.

YOUR OWN LOGO

Why not flash your own logo on the screen while loading SScript, instead of theirs? You can do this as follows:
First, amend line 100 of your 'keys' program to read as follows:

```
printchr$(142):bload"YOUR LOGO",d1,b15,p53248
```

This will ensure that your own logo is loaded instead of SScript's, after loading step 3 above is carried out.
Now prepare your own logo:

Preparing your own logo

Go into Basic, and go into graphics mode (press Norm/Graph key). Draw your own logo on the screen using the control key to access the designs printed on the sides of the non letter keys. The designs on the letter keys are accessed with a Shift key.

Note: Don't touch the ESC key, since this transforms the letters into commands. (See page 30-31 of the gray B Series manual, or page 7 of the Protecto Programmers reference Guide, for these commands.) The Cntrl key also turns some letter keys into commands (see page 6 of the Reference Guide or page 30 of the gray B Series manual).

Note 2: It is useful to cancel scrolling to make sure your precious artwork doesn't suddenly vanish through the screen ceiling by commanding Esc m.

Personally I type uplifting or provocative quotations ('Amateurs Hope, Professionals Work' etc.) surrounded by a jazzy border, but obviously drawing is more fun (for a start why not adapt the 's.enterprise' file in CBUG disk #3 'Swan's Utilities', a neat picture of a spaceship?)

When satisfied with the design, type a command on the bottom of the screen as follows:

```
bsave"YOUR LOGO",d1,b15,p53248 to 55241-120
```

Having done all this, you will have loaded Liz Deal's "pre-superscript" cursor 'fix' program, your own "revised keys" program, and your own logo on the disk in Drive 1 (why not keep this same disk in Drive 1 most of the time, adding essential files to it such as address book, standard letterhead or other printer command headings, and so on, so that you have a ready access data base, word processing shortcuts and also these three SScript loading programs available at all times?). <<Better yet, put all the fixes, standard formats for letters etc, signature and address lines -- put them all on the dictionary disk for SuperSpell. Then when running you only need to start out with two disks and change on -- the program disk to your data disk!>>

Loading SScript then becomes an easy matter of inserting the SScript disk in Drive 0 and following the simple steps 1, 2, 3 above.

<<Tony's programs above are all inthe CBUG library on the CBUG Utilities 2 disk.>>

- -

Report from London visit September 30 1985

Did contact Managing Director John Tranmer of Precision Software, who said they were open to suggesions for improving SScript if we could demonstrate sufficient user interest to make it economical. Brian Leefield Marketing Manager was earnest, insisting that the security lock groaning dosn't hurt the 8050 drives, "it is using a legal command. If the drives suffer, it would be the fault of Commodore". Designer of SSCript is Simon Tranmer, John's brother. Precision Software Inc., 4 Park Terrace, Worcester Park, Surrey KT4 7JZ England. Tel. 01-330-7166 is valid.

Attended annual meeting of Commodore Products Users Group, unfortunately devoted to lenghty reform of Constitution. Most members have C-64s. Only about sixty members have 700's, which is what B's are named there. Chairman Mike Todd has one, and promised to send a memory map of his machine, but hasn't. Jack Cohen is Membership secretary, 30 Brancaster Road, Newbury Park, Ilford, Essex IG2 7BP Tel 01-597-1229. The 8032 expert is Mick Bigwall, tel.953-8385. Their Commodore connection is John Collins, 251-5937, an ex Commodore employee who was demonstrating an Amiga at the meeting. There was a report that a batch of as many as 10,000 Hi Boys are for sale in Germany. <<Does anyone know where/whom these units went to? Are they still in the market?.>>

I did learn that available from JCL Software, tel. 0892-27454, is a compiler for the 700 for 99, and an assembler for 50. Their address is 1 Sheffield rd., South Borough, Tunbridge Wells, Kent TN4 OPD.

For those using the 8432 conversion Supersoft, tel 861-1166, of Winchester House, Canning Road, Wealdstone, Harrow, HA3 7SJ, England. Sells all kinds of 8032 Software, and has a nice little catalogue those interested should request. Included are programming aids, fast sorts, supplies, games. Unfortunately, too many, such as the Chess Program, Grand Master, for $35, involving a plug in ROM which can't be used on Bs.

Sorry, plain forgot to contact Small Systems Engineering Ltd., 2-4 Canfield Place London NW 6-3BT. Tel 328 7145. to check on rumor they make a soft box and hard box and can enable IEEE port to support Corvis hard disk drives.

<<Since Tony's trip to England, brothers Tanmer have developed Superscript III and Superbase II, correcting many of the problems of the present versions not the least of which is reported to be the head banging has been removed. They have decided to agressively support the B128 as confirmed in a phone conversation with John Tanmer 13Feb86. We have also received extensive correspondence from Precision regarding this support and are advised that the upgrades in floppy disk

version will be available thru Progressive Periferals of Denver Co, and in hard disk version thru CSI of Melbourne Fla., all at very economical prices. We believe that Precision is now quite dedicated to the encouragement of B128 usage having recognized the market us CBUGer's represent.>>

QUESTIONS & ANSWERS

by: Warren D. Swan

* * * * * * * * * * * * *

Question: "Can you get the heart, spade, diamond & club graphics in Superscript?"

Answer:

I will start by getting off onto one small tangent that will explain everything; I hope. First, the printer can print very many characters; in fact, nearly 300 different characters when you consider all the upper & lower case letters and digits and special characters ($,@,#, etc.) and punctuation and all the graphics characters (, etc.). Secondly, our computer is limited to being able to handle 256 different possible characters at one time (this is true of all so-called "8 bit" computers). As a result of these 2 facts, the designers of the computer & printer had to make them so that they can be in 1 of 2 "modes" at any given time.

The NORMal mode allows lower case, upper case, digits, special characters and some graphics. The GRAPHics mode allows more graphics, but at the penalty of eliminating the lower case letters. Remember that the computer may be in one mode while the printer is in another. The NORM or GRAPH mode is selected by the key on the computer and by commands for the printer.

So much for that. The reason why SUPERSCRIPT CANNOT HANDLE THE HEART, SPADE, DIAMOND & CLUB GRAPHICS (which answers the question) is that it can only handle the NORMal mode. There is no known way to make Superscript switch the printer to graphics mode and back. If we could, we would be able to get all the graphics characters available on our machine. Sorry!

* * * * * * * * * * * * *

Keywords: Cursor

Question: "From within a BASIC program, how can I turn the cursor on and off without having to use an INPUT statement?"

Answer:

The user may have used an escape sequence to make the cursor not blink or be just an underline, or both. This information is stored in byte 212 (decimal) in bank 15. In order to turn on the cursor and have it be the way the user wanted it, use these statements:

```
  1000 bank 15: poke 55296,10: poke 55297,peek(212)
```

If you want to turn on the cursor and you don't care how the user had it set up, replace the "peek(212)" with a numeric constant as follows:

Number	Cursor	Number	Cursor
0	not flashing; full character size	38	rapid flashing; "underscore" (2 lines at bottom)
6	not flashing; "underscore" (2 lines at bottom)	39	rapid flashing; underscore (bottom line only)
7	not flashing; underscore (bottom line only)	96	normal flashing; full character
24	off	102	normal flashing; "underscore" (2 lines at bottom)
32	rapid flashing; full character	103	normal flashing; underscore (bottom line only)

To turn the cursor back off, use the value 24 in place of the peek(212) above. The above statements will have no appreciable affect when used in direct mode, since BASIC will reset the cursor to whatever is in location 212 after it says "ready." anyway. Besides, the poke 55296,10 must be immediately followed by the poke 55297,whatever. If any intervening statements are executed, or the STOP key is pressed, the effect of the first poke will be cancelled and the second poke will cause unpredictable results to the CRT controller. No problems though.

Incidentally, you can change the cursor "permanently" (ie. until it is next changed) by poking one of the above values into location 212, either from within a program or from within direct mode:

```
    bank 15: poke 212,39
```

(for example) makes the cursor a rapid flashing underscore.

* * * * * * * * * * * * *

Keywords: Variables in Disk Commands, DOPEN, DSAVE, DLOAD, COPY, SCRATCH

Question: "How can I use variables to supply the file name, drive number, etc. in disk commands such as DOPEN, DLOAD, or DSAVE?"

Answer:

BASIC 4.0+ allows you to do that although NONE OF THE MANUALS EXPLAIN HOW. The only reason this author knows how to do so is because I remember reading a review of the 8032 several years back. In the review they discussed the additions in BASIC 4.0 and included examples of how to use variables in disk commands.

BASIC does not care what order you give it the parameters needed for a (BASIC 4.0) disk command. Thus

```
  dopen#1,"data",w,d1,u9
```

is the same as:

```
  dopen w,#1,u9,"data",d1
```

The way it "knows" what is what is by the first character of the parameter, as follows:

PREFIX:	MEANING:	VALID PARAMETER VALUES:
none	file name	valid filename string
#	logical file #	1 to 255
b	Bank	0 to 15

d	Drive number	0 or 1
i	disk ID	2 characters - CANNOT BE A VARIABLE OR EXPRESSION
l	Length of record	1 to 254
p	Physical address	0 to 65535
u	Unit number	8 to 15
w	Write file	none - just indicates write mode instead of read

To use a variable for any of the parameters, start each parameter with its prefix letter, then immediately follow that with a parenthesized variable (or expression). (The disk ID in a header command cannot be a variable or expression, thus i(id$) is not valid). The file name is funny because it doesn't really start with a prefix letter. We could have written the above as:

```
   lf = 1: unit = 9: drive = 1: file$ = "data"
   dopen(file$),#(lf),u(unit),d(drive),w
```

This is true of any of the disk instructions:

```
   fd = 1: ff$ = "old data": td = 0: tf$ = "data": unit = 8
   copy (ff$),d(fd) to (tf$),d(td),u(unit)
```

Again, we could have switched the order of (ff$) and d(fd), and so on. We could have even put the u(unit) before the 'to' instead of after it. Remember that the parameter value may actually be an entire expression, not just a variable:

```
        60000 rem save a copy of the program on the opposite drive with a dot in
        60010 rem place of the 1st character, using replace in case it exists:
        60020 dsave("@."+mid$(prog$,2)),d(1-drive)
```

The above applies to all statements in BASIC 4.0 not found in BASIC 2.0. If you are using the old BASIC 2.0 statements (open, close, load, save, print#, input#, get#), you don't need the parenthesis. The reason is that the parameters given to these instructions must always be in the same order. There is no prefix letter as in the 4.0 instructions. Thus, these are all valid:

```
                                          open lf,device,sa,file$
                                          print#lf
                                          input#lf,a$
                                          get#lf,b$
                                          close lf
                                          load prog$,device
                                          save prog$,device
```

Notice in particular that the number sign is actually part of the INPUT# and PRINT# keywords. So you cannot use PRINT #15 or INPUT #1 because the space causes BASIC to think you meant a normal PRINT or INPUT and the #15 or #1 will cause a syntax error. Nor can you use ?#15 in place of PRINT#15. The number sign may be separated from the GET in a GET# instruction since there isn't a separate "GET#" keyword, only the keyword GET.

* * * * * * * * * * * * *

Keywords: Windows, ESCape Sequences

Question: "How do I use the B's windowing feature?"

Answer:

Your program can set up a window simply by telling the B where the top left character of the window is, and where the bottom right character of the window is. From then on, the B will stay within this new window until you cancel it.

To tell it where the top left character is, move the cursor there and issue an ESCape t. Let's set the top left character at the 3rd line down and 5 characters from the left. From within a program use:

```
     50 es$ = chr$(27)
     1000 print"<2 HOMEs><2 down><4 right)"es$"t"
```

The reason for 2 HOMEs will be explained shortly. For simplicity, we used a variable for chr$(27) (the ESCape character).

We can set the top/left corner directly from the keyboard with:

```
     <2 HOMEs><2 down><4 right> ESC t
```

(Don't enter the spaces. They're there for readability.)

To set the bottom/right corner we use ESCape b. Assuming we have just set the top/left corner as above, we will now set the bottom/right corner at the 10th line down and the 25th character from the left (with respect to the entire screen that is). To do so from within a program we might use:

```
     1010 print"<HOME><7 down><20 right>"es$"b<HOME>";
```

The HOME and semicolon at the end put us back at the top left corner of our little window. Notice that we only used one HOME here, and we only needed 7 cursor downs to get to line 10, since HOME brought us to line 3. Also we only needed 20 cursor rights to get to column 25 because the HOME had brought us to column 5. (Actually, just to confuse things, the machine numbers columns from 0 to 79, instead of 1 to 80, so we're really in column 24, the 25th column. Got it?) We could have done this from the keyboard with:

```
     <HOME><7 down><20 right> ESC b
```

To "get out" of our window we need only issue 2 HOMEs:

```
     2000 print"<2 HOME>";
```

or we could just press HOME twice from the keyboard. Notice that the "set top/left" print statement issued 2 HOMEs in case it had to leave a window that it was already in, before setting up its own.

A single HOME will not cause it to leave the current window. A CLR issued from within a window will only clear the window, not the entire screen. In fact, the window now behaves just as if it were the screen. Cursor up cannot move the cursor above the top of the window, and so on.

Normally we would make the statements that set the upper/left and lower/right corners into a subroutine:

```
     1000 print"<2 HOMEs><2 down><4 right)"chr$(27)"t"
     1010 print"<HOME><7 down><20 right>"es$"b<HOME>";
     1020 return
```

Then we need only gosub 1000 to re-enter this little window. By using several such subroutines, we can have any number of windows. For example, the Real Time Star Trek programs on Warren Swan's disk use some 7 windows. The subroutines to set up each of the windows have line numbers in the 4000s. Each time something is to be presented in a window, the appropriate routine is called first. The overhead is little, since the B is very rapid on display updating (PRINTs). Most of these subroutines leave the cursor at the top/left of the window, as we did above. One of the routines sets up the window and then scrolls it deliberately, leaving the cursor at the bottom left of the window. The result is a "scrolling" window, as opposed to a fixed window.

- - - - - - - - - - - - - - - -

WARREN'S Q & A DODGER

Topic 2: Programmer's Notes

(Each section is continued where it left off, essentially.)

0 PURPOSE:

To get information to you before you have to call or write to a Question & Answer Specialist.

1 DISK:

Assumed: That you are using a Commodore disk unit. All Commodore disk units work in the same manner.

1.4 FILENAME PATTERNS WITH DISK COMMANDS

In the first installment of the Q & A Dodger we saw how filename patterns behave differently depending on the disk command. This is a summary of what files a disk command will act on when wildcard characters (patterns) are used:

COMMAND:	WHAT FILES?	NOTE:	COMMAND:	WHAT FILES?	NOTE:
bload	first matching		dopen	first matching	2
catalog	all matching		load	first matching	
copy		1	open	first matching	2
directory	all matching		scratch	all matching	
dload	first matching		verify	first matching	3

NOTE 1: The only pattern allowed in the "copy" commands is "*". When used after the TO keyword it means "use the same filename here as on the left side of the TO keyword." Thus:

```
copy d1,"&myfile" to d0,"*"
```

will copy "&myfile" from drive 1 to drive 0 keeping the same name.

NOTE 2: Patterns must not be specified when opening a file for output. Although it will not stop you from doing so, the results may be grave.

NOTE 3: The "verify" command treats the filename pattern "*" differently than any others. Immediately after a dsave or save this special pattern means to use the filename just specified. This is useful for saving a file and then using verify"*",8 to see whether it saved with no errors.

The "dsave", "bsave" and "rename" commands should also not be used with a filename pattern. (See note 2 above).

2 COMPUTER:

Assumed: Only that you are using a B Series Computer.

By "computer" I mean the keyboard portion of your system (which is the main unit) in combination with the screen.

2.0 SPECIAL CHARACTERS

Some keys perform an action IMMEDIATELY WHEN TYPED, that is, when INPUT. Other characters perform an action ONLY WHEN "PRINTED" TO THE SCREEN, that is OUTPUT. The strange thing about the latter is that normally when the computer is waiting for input (the next command, or when executing an INPUT statement) the computer "echoes" back the characters so that you can see what you have typed. Thus, while the RVS character only works when OUTPUT, if you press it before typing a command the command will be reversed - black on white. The reason for this is that when you type the RVS key the computer "echoes" it back, thus OUTPUTTING it.

In general, characters that perform their action IMMEDIATELY WHEN TYPED can only perform their action on INPUT. Whereas, characters that perform an action WHEN PRINTED TO THE SCREEN can perform their action whether INPUT or OUTPUT, because the computer "echoes" them.

In the first installment of the Q & A Dodger we had some samples of special characters. Here I will distinguish them a little further for you.

The following characters are acted on WHEN INPUT: function keys (F1 to F20), SHIFT RUN, STOP, the Commodore key, the ESCape key - only if in quotes or insert mode, and the CTRL (control) key. Remember that the ESCape key will get you out of quotes or insert mode, unless you weren't in them, in which case the ESCape will be "passed on" for output processing where it may cause further special functions. The CTRL key actually works on input and output. On input it acts kind of like the SHIFT keys. It is used along with other keys to generate graphics or other special characters. For example, press and hold CTRL and PI to get the "half bright" character. When the B is outputting, holding the CTRL key will cause it so slow down scrolling.

The following characters are acted on WHEN OUTPUT or WHEN ECHOED on input: GRAPH (graphics), SHIFT NORM (normal), SHIFT CLR (clear), HOME, CTRL (as above), RVS, SHIFT OFF, ESCape (followed by a letter), quotes ("), DELete, SHIFT INSert.

In the last installment we mentioned some of the ESCape sequences. Here is the remainder of the ESCape sequences:

ESC b = Set bottom of window. See CBUG Q & A section on windows.
ESC d = Delete the line the cursor is currently on. Subsequent lines move up.
ESC g = Allow beeper to beep (CTRL G, chr$(7) or end of line)
ESC h = Turn off beeper

ESC l = Allow screen to scroll up.
ESC m = Don't allow screen to scroll. Cursor will "wrap" from last line back up to top line.
ESC o = Cancel all modes (insert, quotes, reverse).
ESC p = Erase to beginning of current line.
ESC t = Set top of window. See CBUG Q & A section on windows.
ESC v = Scroll the screen up.
ESC w = Scroll the screen down.
ESC x = Whoops! I didn't mean to use ESC, so ignore it. (Cancel ESC.)
ESC y = Use normal character set.
ESC z = Use alternate character set. Have to make the PROM first.

2.2 HOW TO DISABLE STOP

One way to disable the STOP key is to use the TRAP instruction:

```
  100 trap 3000
       ....
  3000 if (er=14) then resume next
  3010 print err$(er)err$(13)el:stop
```

In this example, the statements after line 100 will execute without fear of the STOP key. If the STOP key is pressed, BASIC will set the built-in variable er to 14 and will switch down to line 3000. Since, in this case, er=14, it will just resume with the statement after the one where the STOP key was detected. If any other error occurs, er will be some value other than 14 and line 3010 will be executed instead. This line simply reports the error and then quits.

There is 2 things wrong with this. The first is because of a BASIC "feature", the second because of a BASIC bug.

If the person using the program presses the STOP key during the executing of most commands, the command is not aborted right away. Instead BASIC waits until the command is finished, and then recognizes the STOP. Input commands (bload, input, input#, etc.), and even some other commands, such as bsave and dsave, will be interrupted by the STOP key before they finish. In these cases the "resume next" will skip to the next instruction without ever completing the interrupted command. This feature can be programmed around by turning the trap on (trap 3000) and off (trap with no line number) as appropriate.

There remains still a bug to thwart our plans. If a GOTO or GOSUB (or ON/GOTO or ON/GOSUB) was being executed when the STOP key was pressed, the "resume next" will go to the statement following the GOTO or GOSUB, rather than to where the GOTO or GOSUB said to transfer to. For example:

```
  200 get c$: if c$="" goto 200
  210 print c$
```

If the user presses STOP just after the 'if c$=""' has been determined to be true, the resume next in line 3000 will transfer control to line 210 anyway, instead of back to 200. I found a stupid way to get around this:

```
  200 get c$: if c$="" goto 200: if c$="" goto 200
  210 print c$
```

Normally the second IF is never executed. In this case it might very well be, since the resume next will send it there. Fortunately it is impossible to press STOP so fast as to cause the second goto to fail also (I tried it many times). However this scheme will not work for GOSUB or ON/GOSUB. Plus it still requires judicious use of turning on and off the trap.

Problems aside, I have successfully used the above schemes to avoid unwanted STOP key interference in some of my programs.

2.3 DISPOSE

The DISPOSE instructions solves a problem that has always existed in Commodore BASIC (as well as other Microsoft BASICs). The DISPOSE instruction does one of two things, depending upon what follows the word dispose.

DISPOSE FOR is used to toss away the currently active FOR/NEXT loop. If, for example, we want to find out how many elements of the array b are already in the array a we could use:

```
 1000 for b=1 to nb: rem for each element in array b up to nb elements
 1010 for a=1 to na: rem check through the na elements of array a
 1020 if b(b)=a(a) then n = n + 1: go to 1040
 1030 next
 1040 next b
```

Line 1020 goes to line 1040 after counting the match, because we don't want to see HOW MANY TIMES an element in b is in a, but rather IF it is. When this happens, BASIC sees the "next b" and decides that we must have finished the inner loop (for a/next), and now want to do the next iteration of the "for b/next" loop. Had we left off the "b" in this next statement, BASIC would have assumed that we were still continuing the inner loop.

In this case we had no trouble. However, there are times when leaving a loop unfinished can cause trouble later on. A better way to program the above would be:

```
 1000 for b=1 to nb
 1010 for a=1 to na
 1020 if b(b) = a(a) then n = n + 1: dispose for: go to 1040
 1030 next
 1040 next
```

The "dispose for" tells BASIC explicitly that we want to leave that inner loop. You can use more than one "dispose for" to exit several levels of for/next loops. Notice that we don't need the b after the next in line 1040, since BASIC now knows that we can only be talking about the for b loop when line 1040 is executed. Either the inner loop finished (element b(b) not found in array a), or the dispose for "got rid of" the inner loop.

The other use of dispose is to exit a subroutine in such a way that the routine that GOSUBed to it will also be "aborted." For an example, we are writing a main program that is to process some data. It will use a subroutine at line 2000 to read in the next set of data (several lines):

```
  10 rem main program
   ...
  230 gosub 2000: rem read section of data
  240 if e<>0 goto 9999: rem check for errors encountered in subroutine
```

```
     ...
```

As a matter of fact, this subroutine at 2000 is going to use another subroutine at 3000 which will take care of reading one line of data:

```
     2000 for l=1 to 20
     2010 gosub 3000: rem read a line of data
     2020 if e<>0 then return: rem if errors, let main program recover
     2030 rem if end of file reached before 20 lines read, return with error:
     2040 if l<20 and (st and 64)<>0 then e = e+1: return
     2050 next
     ...
     2100 return
```

The subroutine at 3000 reads the lines and checks for certain errors:

```
     3000 input#3,data$
     ...
     3090 if (some error encountered) then e=e+1
     3100 return
```

The DISPOSE GOSUB command allows us to simplify this a little. First, eliminate line 2020 where the first subroutine checks for errors from the second subroutine. Then modify the second subroutine (at line 3000) as follows:

```
     3000 input#3,data$
     ...
     3090 if (some error encountered) then e=e+1: dispose gosub
     3100 return
```

The trick here is that the return in line 3100 does NOT return to the statement following the gosub 3000 in line 2010. Instead it returns all the way back to the main program in line 240. The routine that handles 20 lines of data does not care whether the routine that reads one line encounters an error. If an error did occur, the latter will return all the way back to the main program with the e variable set to indicate an error. DISPOSE GOSUB may be executed repeatedly to cause the next RETURN to return back several levels of GOSUBs.

DISPOSE FOR is very valuable and recommended. Use of DISPOSE GOSUB is very handy in some situations, but can become obscure if abused.

3 PRINTER:

Assumed: That you are using a Commodore 4023 or like Commodore dot-matrix printer. Also, that the printer's device number is 4. If not, replace 4 with the appropriate device number in the following.

3.5 CHANGING THE NUMBER OF LINES PER PAGE

The Commodore printers are strange in that they require special "open" statements in order to change how the printer works. Most other printers only require that special characters be sent to them to change how they work.

If you are using special forms, you will have to be able to tell the printer to change the number of lines per page. This is only true if you are going to use the printer's built-in pagination mode. Most pre-packaged software simulates pagination, rather than having the printer do it. However, to get listings of programs, you still need the printer's pagination mode.

When the printer is powered up, it assumes that there are 60 printable lines per page in pagination mode. 3 lines are skipped at the top and bottom, so that the total is 66 (6 lines per inch times 11 inches). As far as pagination goes, the printer only cares about the number of printable lines per page. If we switch to a 14 inch form, we now have 14 inches times 6 lines/inch (= 84 lines) minus the 6 inch spacing, equals 78 lines per page. To tell the printer this we execute:

```
   open 3,4,3: print#3,chr$(78): close 3
```

The printer knows we want 78 lines/page now, but it will not actually put it into effect until the next page begins. We can force that by sending the HOME/CLR sequence to the printer. The 4023 can handle down to 14 lines/page and up to 127 lines/page.

3.6 CHANGING THE NUMBER OF LINES PER INCH

Another nice feature of the Commodore printers is the ability to change the number of lines per inch. By doing so, you can actually have your printer print contiguous graphic pictures.

Unfortunately the printer manual is totally inaccurate here. Some of the Commodore printers had line feed motors that could handle up to 144 steps per inch. Other Commodore printers (such as the 4023) could handle up to 216 steps per inch. You will notice that these mystical numbers actually come from the idea that we want to be able to choose so many of these little steps to make up so many lines per inch. In order to figure out how many steps are needed to achieve LPI lines per inch, we would divide, as appropriate:

```
      steps = 144 / lpi: rem for some Commodore printers
      steps = 216 / lpi: rem for other Commodore printers, such as the 4023.
```

The numbers 144 and 216 were chosen because they can be divided into so many different numbers. The reason for this is that

```
      144 = 2 * 2 * 2 * 2 * 3 * 3, and
      216 = 2 * 2 * 2 * 3 * 3 * 3.
```

OK? Unfortunately, Commodore got mixed up about which printers had which number of steps per inch. Fortunately, all we have to do is try some examples and get out our ruler and measure.

When the printer is turned on, it is set to 6 lines per inch. That is 216/6 or 36 steps (not the 24 that the book states). To change it to 8 lines per inch, we would first calculate 216/8 = 27, then issue these commands:

```
      open 6,4,6: print#6,chr$(27): close 6
```

We could have let BASIC do the calculation by simply using chr$(216/8). Notice that the printer really can't do exactly 5 or 7 lines per inch because 5 and 7 don't go into 216 (or 144) evenly. But it will get pretty close.

Another thing that changed over time is the size of the characters printed by the printers. With the older Commodore printers, 8 lines/inch would allow each line of characters to be contiguous for drawing pictures or making charts or what have you. The 4023 characters are smaller and require 12 lines/inch in order to get contiguous lines. So we would have to use:

```
     open 6,4,6: print#6,chr$(216/12): close 6: rem 216/12 = 18 steps.
```

- -

<<And now, folks, Warren in his hurry to get his last Q&A column to us neglected t provide adequate information on his library disk, so here it is. Remember, only if you support our contributing members by freeware gratuities and by ordering their sometimes royalty bearing disks will we enjoy the aid of these superior sources of help. Warren Swan has put in many a sleepless night keeping up with member correspondence, writing his course in basic 4.0+ ala B128 and preparing materials for future release. Such a sacrifice need be rewarded. Say thank you with your purchases and gratuties! <<Order Warren's prior disk, #12881 at $14.00. You won't be disapointed.>>

" start up"   First program on disk. Contains brief info on others.

ARTICLES:
"& file prefixes" Why and how to use file prefixes to organize files.
"& cursor config." How to make your cursor fancy or plain.
"& program lines" A look at how BASIC stores program lines

INSTRUCTIONS:
Complete instruction files for programs below. Each comes in a Superscript text file format so you can print it the way you like it, and a post-Superscripted form in case you don't have Superscript, or just want to read the instructions quick without using Superscript.

GAMES:
" star trek 2"   Two versions of this action packed Star Trek game
" star trek 1"   Uses B's windowing feature so you can really see what's happening while you play. Also uses the built-in clock so that you actually play against real time
" eliza"   Have a session with a wacky psychiatrist - expandable
" iq test"   Uses statistics to approximate your (math) I.Q.
" craps"   Welcome to Lost Pesos. Shoot a few rounds - with grap'

DEMOS / MATHEMATICS:
" lifetime"   Determines your life expectancy based on actuarial info.
" calculator"   Use your B as a big RPN type calculator
" factors"   Determines factors of a number
" regression"   Linear, exponential, power, and logarithmic regressions
" squirm"   Wild

UTILITIES
"!futility"   Actually 5 programs in one. (1) Recover scratched files. (2) Count characters, lines and blocks in a file. (3) Selectively scratch files from a disk. (4) Selectively copy files from drive to drive. (5) Directory a disk. All work on any drive unit #.
"!copy"   General device to device copier. Keyboard to RS-232, RS-232 to disk, disk to screen - you name it, it does it
" rename disk"   Allows you to rename any disk
" archive 1"   Archive many files into 1 file (for real!)
".get filenames"   General purpose "pattern to list of files" subroutine
".get mach lang"   Subroutine to search for & load M/L or any BLOADable

GRAPHICS:
" large letter"   Allows you to type large messages on the screen. 5 rows of 19 characters, each approximately 1" by .75"
" calendar"   Perpetual calendar (from 1752 on). Display any month on screen. Or find day of week given date.
" s-enterprise"   Displays a graphic Enterprise on the screen - with "motion"
" p-enterprise"   Prints an Enterprise on CBM 4023 Dot matrix printer.
" bouncing ball"   Also wild!

COMMUNICATIONS:
" terminal"   Program to allow your B to act like a terminal. Fully configurable and down/up loads to/from disk. Includes complete ascii character set such as necessary for mainframe programming.

THE B128 AND THE DEVIL
or
INTERFACING WITH DIABLO

The Diablo HyTerm 1610 with HyType II Printer Model 1345A
<<NOTE: This printer is also available under the Xerox trademark as a terminal.>>
Warren Kernaghan, Kansas City, Mo.

This printer is now old enough to be appearing in various ads offering used equipment at reasonabl rices. These are extremely well built printers, weighing 40-50 pounds, and are intended for heavy duty use.

The one I have was built about 1978, looks almost like new with very little wear, and is capable of 110, 150, 300, or 1200 baud connection. It will print up to 45 characters per second (cps). An RS-232-C interface is standard, with 20 or 60 mA current loop interface optional (to simulate a Teletype).

Print wheels are available in 10, 12, and 15 characters per inch (cpi). Pryor's catalog lists 7 choices available for about $6 each. Eleck-Tech also shows them in their catalog for about the same price. The same print wheels are used in various models of Canon, C. Itoh, DEC, Hewlett Packard, Texas Instruments, and Xerox. Don't expect to find them at your corner typewriter store, though. Ribbons are the same as many later models and cost about $3.25 up to $8 depending upon source and brand.

Interfacing gave me more trouble than I care to admit, but it is typing away very nicely now, as our first quality printer. It should last as long as we do.

Initially, the printer RS-232-C cable connector was wired as if it were a computer, the same as the B128:

1	Protective ground		4	Ready to send	(RTS)	7	Signal ground	
2	Transmitted data	(TXD)	5	Clear to send	(CTS)	8	Data carrier detect	(DCD)
3	Received data	(RXD)	6	Data set ready	(DSR)	20	Data terminal ready	(DTR)

After seeing that the pin connections of both computer and printer were the same, it seemed obvious to me that making a "null modem" cable would solve the whole interface problem. For those not familiar with "null modems", they consist of a pair of RS-232 connectors with or without a cable between them, and are used to connect two devices wired the same, as in this case. The two connectors are wired as shown here:

1		-	1		4	RTS	-	5	CTS	7		-	7	
2	TXD	-	3	RXD	5	CTS	-	4	RTS	8		-	8	
3	RXD	-	2	TXD	6	DSR	-	0	DTR	20	DTR	-	6	DSR

You can see that 2 and 3 are reversed, so that transmitted data from the computer ıs fed into the received data pin on the printer. Anything transmitted from the printer to the computer (its transmitted data, TXD) is fed to the computer's received data connection (RXD). Likewise, the computer's ready to send (RTS) signal should go to the printer's clear to send (CTS) connection, as should the remaining handshaking connections on pins 6 and 20.

When either of these connectors is plugged into a computer wired as ours is, and the other is plugged into a printer or modem, the odds are good that all will function correctly when baud rate, number of bits, parity, and stop bits are also correct.

Naturally, I wanted the top speed of 45 cps, so the printer was set for 1200 baud, and the usual 8 bit word length, with 1 stop bit, and no parity.

What I got, even after much experimentation with word size, stop bits, parity, and baud rate, was great frustration and at most a few satisfactorily printed lines before the error light on the printer came on and the garbling began. Also an error on Superscript. No combination of these could be found that would improve matters noticeably.

Exhaustive perusal of the manual that very fortunately came with the printer brought to light the information that "a simple ETX/ACK protocol is built into the microprocessor program which can be used to guarantee no data loss...", and another comment, "when the 1200 baud mode is used, the ETX/ACK protocol should be employed to prevent data overrun". Hmmm. A little research confirmed my worst suspicions, i.e.: ETX/ACK protocol is a software handshake not, in this case at least, compatible with the RS-232-C handshaking.

The whole 40 lbs. and associated pages of notes on attempts to interface the beast were consigned to a heavy duty shelf in another room while I polled friends who might have ideas on solving the dilemma. No worthwhile suggestions were found locally, (I don't know everyone!), so a call was made to Norman Deltzke, who provided a list of possible contacts. Fortunately, while waiting for evening to arrive, when the various consultants would be available, last ditch perusal of the manual turned up a previously passed over paragraph on preventing buffer overrun. Briefly, it mentioned that "pad" characters such as NULs or DELs could be used after carriage returns, tabs, etc. to prevent buffer overrun that could occur at higher print speeds.

That was convincing enough for me to carry the printer back out to the computer again for a few more tries at taming the thing. Even with Superscript and Diablo set to 110 baud, (10 characters per second), no improvement was noted. There had to be a reason other than software handshaking and buffer overflow that the printer wasn't operating properly.

Another check was made on the RS-232 pin voltages, with and without printing in progress. The only possibility of a problem was found at pin 8, the data carrier detect (DCD) line. It measured a few tenths of a volt positive no matter what was taking place. I knew that the B128 didn't care if pin 8 was connected to ground or a positive voltage, so hadn't been concerned about it. Also, the Diablo manual said that the DCD line wasn't sampled by the printer. On the chance that it might help, pin 8 was connected to pin 11 at the computer end, which is nominally +5 volts.

One more test at 110 baud, and EUREKA! The printer chugged along slowly for a whole page with no errors. The baud rate was changed to 150 baud, then to 300 baud, where it remains. For a neater system, the null modem cable was replaced by re-wiring the RS-232 connector on the printer cable. Even if it does have an 8080 microprocessor and software handshaking capable of 45 cps, I don't think it worthwhile writing a program to read Superscript files and handle the software handshaking. For the amount of printing we do, 30 cps of quality printing for a little over $100 is quite satisfactory. However, if someone has an easy solution, that's another matter.

With a letter quality printer, three quick taps of the return key when the Superscript menu appears will get you nowhere. Text width is best left at 80 columns until ready to print, in my opinion. Printer type for Diablo will be No. 3. When asked ascii or cbm, choose the default (a) by pressing return, then choose (r) for RS-232. You are asked for settings for the Control and Command Registers. The correct value for the Control register is 6 (300 baud, 8-bits, 1 stop bit), and for the Command Register is 0 (parity disabled).

On the printer, set the speed switch to 30 for 30 cps, and the parity switch to mark. Parity isn't checked in that position. Duplex should be full. Auto LF should be on. Form length normally will be set to 11. There is a place on one of the PC boards for a jumper when 1200 baud is used. This jumper should be removed. The board will be at the left front when the main cover is removed. Board labelling will be HPR01 or HPR02.

Superscript controls listed for the later model Diablos work fine with this one: underlining, bold printing (strikes 3 times), shadow printing (even bolder), superscript and subscript, 4-12 lines per inch, 8-15 characters per inch, backspace, and quite a few more using ESC codes.

To reset the printed width after typing your text at the 80 column default width, press ESC shift CLR. After answering 'yes' on query, the width can be set as desired. I find 130 suits most of my needs. Pick whatever seems appropriate and test with the printer. It is useful to print a 40 character line with all possible vertical and horizontal spacings and printwheels. The resulting display will greatly aid in making choices of settings later without continual retesting. Another table of tested printer controls and escape codes is also invaluable for handy reference, at least until you learn them all by experience. A plastic holder is one way of preserving them, or possibly laminating them back to back.

Should you be interested in one of the old Diablos, the Computer Shopper magazine is a good one to check, as well as various computer resale stores, surplus centers, etc. BNF Enterprises in Peabody, MA (617) 531-5774, has been offering Diablo HyType model 1355WP printers used, untested, for $105 each. You would be well advised to stay away from the older Hytype I printers because of replacement parts and supplies problems. Series 1200 are HyType I, while series 1300, including the 1345 and 1355, are HyType II. Manuals are available from Diablo Systems, Inc., Retail Store, 1231 Columbus Drive, Richardson, TX 75081. Phone: 800 527-4614. Also in San Jose, CA 800 351-9300. Parts are available through Xerox service centers.

If you want to contact me about Diablos, don't ask about any models other than mine. I'm only familiar with this one.

For those concerned about using a modem and printer connected to the same port, disconnecting and connecting to shift back and forth: You don't need to switch all the wires in the cables. I used a surplus 6-pole 2-position rotary switch, and connected 2, 3, 4, 5, 6, and 20. Small good looking cabinets about 4 inches cubed are easy to find at places like Radio Shack, and RS-232 connectors are available new and used. You can also buy switch boxes complete for RS-232 for about $125 if you prefer that route.

- - - - - - - - - - - - - - - -

A letter from Warren Kernaghan, 16 Nov 1985

To: CBUG & Lawrence Cooper:

Your article "Unscrambling the B-128 RS-232C Port" on CBUG's Prerelease #1 finally gave me the boost needed to tackle a nagging problem present since purchasing a Signalman Mark XII modem many months ago.

The first problem was that it absolutely did nothing. A friend who had one found that the DCD line could not remain directly connected to the port, but could be reconnected after start-up was completed.

The Programmers Reference Guide, page 124, states that DCD high logic means that the carrier signal is present, and low logic means it is not. It also states that it must be driven, meaning high or low, but not switched (or presumably, floating), and that DCD must be low for the receiver to operate.

Based on all that, I lifted the conductor at pin #8 on the modem connector and installed a wire from #8 pin to a ground screw on the computer. Operation seemed OK, except that the modem wouldn't respond to command signals unless the computer had been on for about 15 minutes. Also, few result codes were displayed (unless the telephone handset was off-hook!) Thinking the problem was in the modem, I returned it for service under warranty. It came back with a note that no problem was found.

Your article appeared after I'd lived with that for some time, and got my interest up, so the first thing done was to check voltage levels cold and warm to spot any changes during the first 15 minutes.

No significant changes were found, so comparisons were made between these levels and those you mentioned. The first surprise was that the nominal +12 VDC was more like 10.5 - 11 VDC. That seemed reasonable after checking the schematic diagram.

The first major breakthrough was in noting that the DCD line should be +12 VDC rather than at logic 0. That seems to contradict the manual, but it is certainly a true statement.

When the DCD line floats, the modem responds immediately, even with a cold computer. It does the same with +4 volts or +11 volts. As there is no +12 VDC present at the RS-232 connector, I placed a jumper in the modem connector between pins 11 & 8. This is nominally +5 VDC, actually +4.4 VDC because of a series connected silicon protective diode. Could this indeterminate logic level prove too low for a logic line expecting something closer to +12 VDC? The 1489 gates are supplied only with +5 VDC, so I think the +4 VDC appears to them as a solid logic 1.

Everything is operating better than ever before, thanks to impetus your article provided.

There must be others with various problems and questions regarding modems as applied to the B-128 computers. Perhaps CBUG would publish information on the modems providing satisfactory operation as connected, those working after minor wiring changes - as the Mark XII did, and those not performing satisfactorily as yet. I'd vote for more information!

Thanks for your article...

Warren

INTERFACE A TYPEWRITER TO THE B-128??? SURE YOU CAN!!!

by: Beverly A. Rumsower

I belong to a dog obedience training club and publish a monthly newsletter for that organization. The newsletter is 8 to 10 pages in length - every month, month after month, year after year (for the past seven years). This meant that every month around the 15th, I'd sit down with my ever faithful IBM Selectric and spend at least two evenings (sometimes more) and maybe even a couple of mornings before going to work, typing, typing, retyping ... because this paragraph really didn't belong in that particular place ... or it shouldn't even be there at all, etc., etc. This was my whole reason for getting a computer - I was tired of reams of paper ending up in the trash. Now, please remember that I am a 51 year old grandmother and that up until two years ago, I had never even touched a computer, was scared to death of the machines and thought they were monsters who could eat you alive!!

About the same time that I became the proud owner of my B-128, a newsletter came due. My ever faithful IBM Selectric, knowing its days were doomed, became very ill. The typewriter repair people said that it was terminal, hopeless and that nothing could be done to save my faithful friend ... it had died. Not having time to really learn how to use the B-128 (my newsletter was due in 3 days), I bought a Smith Corona Memory Correct III electric typewriter (on sale for about $300). This got me out of the woods as far as that particular newsletter was concerned. Before the next newsletter was due, good old Superscript II and I became very fast friends and the new typewriter was covered up and shoved to the back of the table.

Two or three months passed and one day I needed to type something that was wider than 80 columns (as you know the 4023 won't do anything wider than 80 columns). Getting out my typewriter manual to check margin settings, etc, I discovered that it made reference to an RS232 interface cable. Wait a minute, I said to myself, doesn't my computer manual make reference to the same cable??? So, out comes the computer manual and YES ... there is the same RS232 cable mentioned! Was I ever excited ... WOW! ... Could it be possible that my lovely new Daisy Wheel Typewriter could become an extension of my computer?

I called the place where I bought the typewriter and was not given much encouragement ... sure, I could buy the "messenger module" for the typewriter and they could have it installed for me (at a very high cost) but the minute I mentioned B-128, no one seemed to know whether it would work or not. The attitude was to go buy the industry standard RS232 cable and plug it in. If it worked, fine...if it didn't, that was fine too. Well, I don't have money to just throw away experimenting so I called Interstate Office Machines here in Kansas City. They are an authorized Smith Corona Repair Center and the place where the typewriter store was going to send the typewriter to have the messenger module installed. Bill Macy is the owner and has worked for Smith Corona. Bill was more than helpful. After explaining what I wanted to do, Bill called Smith Corona in New York and talked with their engineering department. They told him what had to be done to the inside of the typewriter as far as installing the Messenger Module and then gave him the instructions to give to me on how to make my own RS232 interface cable for the B-128. (It is my understanding that there are some B-128 machines in the engineering department at the Smith Corona factory.)

While Bill Macy was installing the messenger module on the typewriter, off to Radio Shack I went to purchase 10' of regular 4 wire telephone cable, and 2 standard 25 pin plugs for the RS232 port. My son-in-law did the soldering for me. According to Smith Corona, this is the way it had to be done:

Computer Plug /	Typewriter Plug	
pin 2	pin 3	red wire
pin 5	pin 4	yellow wire
pin 7	pin 7	green wire
pin 1	pin 1	black wire

When you boot up Superscript, set your margin (default setting for 80 columns or 240 for wide paper), default for the disk and choose 4 for the printer. A screen will come up asking if you want (a)scii or (c)bm Code ... choose (a). Your next choice is (i)eee, (r)s232 or (c)centronics. Choose (r). The next screen shows RS232 Definition. The Control Register Value(0-255) needs to be set at 8 and the Command Register Value (0-255) needs to be set at 96. Press return and you are in the edit mode for Superscript!

When you are ready to have the typewriter work as your printer, press the "Code" button in the lower left hand corner of the typewriter and the letter "P" at the same time. The messenger module red light will come on and your Smith Corona typewriter is now a printer for your computer. Select your pitch (10, 12 or 15) before you turn on the messenger module. All of your margin settings can be done through the computer just as you would do for your 4023 dot matrix printer. You will need to experiment with paper placement in the typewriter to have it print where you want it. The typewriter will now print approximately 200 words per minute which is a lot faster than I can type!

If there is any further information that I can give, please let me know. My address is: 1800 East 84th. Terrace, Kansas City, MO 64132. Please enclose a stamped, self-addressed envelope if requesting information.

GENERAL DATA ON THE RS 232 BUS

by: Charles Ames

There are two main types of RS 232 connections The first is "data-terminal-equipment" to "data-terminal-equipment" which is used when connecting two devices which will communicate "back and forth". Examples of this are connecting one computer to another computer or when a computer is connected to a device such as a printer or a disk drive. Shown in figure 1. is the normal "data-terminal" to "data-terminal" connection.

```
pin              to                 pin
 1 --------------shield
 2 ---------------->------------------ 3
 3 ----------------<------------------ 2
 4 ---------------->------------------ 5
 5 ----------------<------------------ 4
 6 ----------------<------------------ 20
 7 ---------------<->----------------- 7
 8 ---------------<->----------------- 8
20 ---------------->------------------ 6
```

FIGURE 1.

```
pin              to                 pin
 1 --------------shield
 2 ---------------->------------------ 3
 3 ----------------<------------------ 2
 7 ---------------<->----------------- 7
```

FIGURE 2.

This method is for a full hardware handshake. Another method for "data-terminal-equipment" to "data-terminal-equipment" which assumes no hardware handshake is shown in figure 2.

The other type of RS 232 connection is the "data-terminal-equipment" to "data-communication-equipment" which is used when connecting a device, such as a computer, to a modem. Below in figure 3. is the normal "data-terminal-equipment" to "data-communication-equipment" connection.

```
pin              to                 pin
 1 ------------ -shield
 2 ----------------->------------------ 2
 3 -----------------<------------------ 3
 4 ----------------->------------------ 4
 5 -----------------<------------------ 5
 6 -----------------<------------------ 6
 7 -----------------<->---------------- 7
 8 -----------------<------------------ 8
20 ----------------->------------------ 20
```

FIGURE 3.

```
pin              to                 pin
 1 ----------------shield
 2 ----------------->------------------ 2
 3 -----------------<------------------ 3
 7 -----------------<->---------------- 7
```

FIGURE 4.

The three-wire, "data-terminal" to "data-communication" with no hardware handshake is shown below in figure 4.

THE COMMODORE RS 232 OPEN STATEMENT

The Commodore B128 RS 232 open command has the format, open "file number",2,"secondary address","open-string" where:

1. "File number"=Logical file number to be used with the channel.
2. "Secondary address"=The type of character conversions which the Commodore will perform. This is usually set at 131.
3. "Open-string"=A four byte code of which the first two bytes determine the parameters of the RS 232 channel.

The first byte of "open-string" will determine the baud rate or data transfer rate, the word length, and the number of stop bits. These three parameters must be set the same for both devices. To find the correct value for byte one of openstring, find the baud rate, word length, and number of stop bits that the device to be connected to the Commodore is set for then find the match in the corresponding tables below and obtain a value. These three values added together plus 16 will give the value of byte one of "open-string".

BAUD RATE	VALUE	WORD LENGTH	VALUE	# STOP BITS	VALUE	BAUD RATE	VALUE
50	1	8	0	1	0	1200	8
75	2	7	32	any but 1	128	1800	9
110	3	6	64			2400	10
134.5	4	5	96			3600	11
150	5					4800	12
300	6					7200	13
600	7					9600	14

When in doubt on the number of stop bits use a value of 128.

The second byte of "open-string" will determine the handshake, duplex, and parity used. To find byte two, determine the type of parity that the device to be connected to the Commodore is set for and then get a value from the table below. Add 1 if you are using a "data-terminal" to "dataterminal" connection. If you are using a "data-terminal" to "data-communication" connection add 16.

PARITY TYPE	VALUE	PARITY TYPE	VALUE
none	0	mark	160
odd	32	space	224
even	96		

The third and forth bytes of "open-string" are not used but must be included as "++" on the end of "open-string". Below are several examples of actual open statements followed by brief explanations.

```
open 1,2,131,chr$(8+16+128)+chr$(32+1)+"++"
```

Opens file #1 an input/output channel that converts from cbm to ascii. Baud rate-1200, internal clock, 8-bit word, more than 1 stop bit. Odd parity, "data-terminal" to "data-terminal".

```
open 4,2,131,chr$(10+16+32)+chr$(96+1)+"++"
```

Opens file #4 an input/output channel that converts from cbm to ascii. Baud rate-2400, internal clock, 7-bit word, 1 stop-bit. Even parity, "data-terminal" to "data-terminal".

CONNECTING THE B128 TO THE MICRONICS STAR DELTA-15

The Delta-15 was connected to the B128 using the "data-terminal-equipment" to "data-terminal-equipment" diagram shown in figure 1. This is called a "null-modem" cable and has wire pairs 2/3, 4/5, and 6/20 reversed.

The Delta-15 has three banks of DIP switches which must be set. DIP bank one has 8 switches, DIP bank two has 4 switches, and DIP bank three has 8 switches. Bank two is the only bank which can be set without removing the printer cover. It is located on the lower, right-hand, rear of the printer. Bank one is located directly behind bank two. Bank three is located on the second-level printed circuit board. The set-up we used is shown in figure 5. Only those settings shown in parenthesis () should effect the RS 232

	DIP BANK ONE	DIP BANK TWO	DIP BANK THREE
SWITCH 1	ON	ON	(ON)
SWITCH 2	ON	(ON)	(ON)
SWITCH 3	ON	ON	(ON)
SWITCH 4	ON	ON	(ON)
SWITCH 5	ON		(ON)
SWITCH 6	ON		(ON)
SWITCH 7	ON		(OFF)
SWITCH 8	ON		(OFF)

FIGURE 5

This sets the printer for 7 data bits, parity checked, xon/xoff, odd parity, baud rate-1200

The open statement used for the Commodore is:

```
open 1,2,131,chr$(8+32+128+16)+chr$(32+1)+"++"
```

This sets the Commodore for baud rate 1200, 7 data bits, stop bits-(the printer manual was not specific, so 128 is chosen) ,plus 16.

Parity odd, and since we were using a "data terminal" to "data terminal" connection, add 1.

NOTES ON USING THE RS 232

1. The (chr$) codes between 64 and 91 do not work when sending commands to the printer. Substitute the appropriate letter from the appendix at the end of the Commodore users guide. examples:

```
print#1,chr$(65)            becomes          print#1,"A"
print#1,chr$(76)            becomes          print#1,"L"
print#1,chr$(69)chr$(80)    becomes          print#1,"E""p"
```

note that the letters are CAPITAL letters.

2. When connecting a device to the B128, start out with a slow baud rate to avoid buffer problems. Remember baud rate for both devices must match.
3. If communication between devices is non-existent, sluggish, or parts of the data is missing, ect., recheck settings on both devices.
4. If there seems to be an "echo" in the data. Or on start-up the device seems to run wild, try adding 16 to byte two of open-string.
5. If all else fails experiment with different open statements. Changing one thing at a time and observing the results to give a clue as to what is happening.
6. This was written using SUPERSCRIPT II and a delta-15 printer hooked-up to the RS 232 port. The session was started by entering:

A. text width--------------80 columns
B. disk---------------------device 8
C. type of printer---------3. Qume/Diablo/8300/6400
D. printer interface-------ascii
E. port--------------------RS 232
F. control register value--184
G. command register value--33

7. Using Figure 1. connections and an expanded version of "prog3" we were able to interface the B128 to a data-aquisition system which would, on command from the Commodore, take 1000 voltage readings in two seconds and then transfer that data back to the B128 to be processed, stored on disk, and then printed in graphical form. The program in the data-aquisition system was entered and debugged from the Commodore keyboard.

SUMMARY OF DISK PROGRAMS

<<re related programs in CBUG library>>

1. "prog1": Demonstrates how text can be altered on the Delta-15. Shows how commands are printed to the RS 232 port.
2. "prog2": Defines and prints enlarged numerals (0-9) over the RS 232 port using the "cmd" statement.
3. "rs 232 bus" and "rs 232 bus a": Discriptions and data on the RS 232 bus in SUPERSCRIPT II form. Both programs are the same.
4. "prog3": Short program which will print (to the screen) all characters it receives over the RS 232 bus and will print to the RS 232 bus data typed from the keyboard. Used to test and trouble-shoot RS 232 connections. this program can be expanded to do many things. One example is to use the Commodore as a terminal to enter programs into another computer.

POWERTYPE DIP-PIN

by: William H. Wolf

settings are these:

DIP-#1; 1 = ON; 2 to 7 = OFF; 8 to 10 = ON ---- DIP-#2; 1 to 5 = OFF; 6 = ON; 7 = OFF; 8 & 9 = ON; 10 = OFF

PAPER = 11 SLEW = cr/lf PITCH = 12 LINE = 6 COPY = 1 STATUS = PM

COMMODORE SETTINGS ON POWER-UP OF SUPERSCRIPT:
PRINTER #4 ASCII RS 232 C CONTROL REGISTER 8 COMMAND REGISTER 0

COMMODORE BASIC OPENING OF RS232C CHANNEL: OPEN 3,2,131,chr$(8);chr$(0);chr$(0);chr$(0)
(see pages 137-140 of the Commodore Manual)

RS-232-C PINOUTS:

COMMODORE	POWERTYPE	COMMODORE	POWERTYPE
1	1	5	20
2	3	7	7
3	2		

(ALL OTHERS ARE NO CONNECTION)

NOTE: Any made-up RS232 cable should be twisted pairs and 36 inches or less in length to reduce capacitive interaction which limits the BAUD rate.

William H. Wolf, 1690 Hosmer Road, Appleton, NY 14008

RS-232 INTERFACE FOR COMREX COMRITER-IIE LETTER QUALITY PRINTER

by: Ron Hohs

The Comrex printer is a daisywheel letter quality printer operating at 12 CPS, with a bidirectional carriage and an intergral 3K buffer. Tractor feed or cut sheet feeder is available as an option. It is available from ELEK-TEK inc. 6557 N. Lincoln Ave. Chicago Il. (312) 677-7660 or (800) 621 1269.

The ComRiter-IIE interfaces to the b-128 with the following cable conections:

B-128		Printer
1	<---------------------->	1
2	<---------------------->	3
3	<---------------------->	2
4	<---------------------->	5
5	<---------------------->	4 & 8
6	<---------------------->	20
7	<---------------------->	7
20	<---------------------->	6

When using this printer with Superscript select printer type (4) from the first menu. The next screen will prompt you for the code type and interface type. Specify ascii code and rs232. Yet another screen will come up asking to specify the rs232 definition. The printer can be configured to the rs232 default values thereby eliminating the need to remember the values to enter for the control registers. In this way only a return is needed to load the register values.

To set the printer to the default values for rs232 the dip switches on the back of the printer need to be set. Dip sw1 (first switch to right of rs232 connector) has all positions closed (down). Dip sw2 (farthest right from the rs232 connector) has positions 1,3,4,6,7 open (up) all others are closed (down).

When running Superbase the printer can be accessed by entering the following command: "pdef 2:pdev 2,8,0" (do not enter the quotes). This can be entered directly on the command line or it may be incorporated into a report program.

UNSCRAMBLING THE B-128 RS-232C PORT

by: Lawrence D. Cooper
604 Cutters Mill Lane, Schaumburg, Il. 60194

GENERAL BACKGROUND INFORMATION

The RS-232C standard interface is a means of controlling the transmission of serial data by computers. When implemented exactly according to the standards set by the E.I.A., it defines 25 different wire connections that perform unique and essential functions in the preparation, start up, transmission, and shut down of a computer serial port. In addition to the definition of specific wire connections with their functions, the standard also recomends a special connector. To the credit of the designers of the Commodore B-128, the connector on the rear of the B-128 that is labeled as 'RS-232C' is the correct type of connector for all standard RS-232C uses in the United States. (We will call this connector the DB-25 connector from now on.)

In addition to the physical and theoretical definitions of wire connections, wire functions, and connector to be used, the E.I.A. also set down a standard for how each of these wire connections should interact during the process of preparation, start up, transmission, and shut down of data. This interation is most commonly refered to as handshaking. There are different kinds of handshaking for modems, terminals, computer ports, etc.

From all of this, it should now be pretty apparent that getting two serial devices to talk to one another is not just a matter of plugging them together. From what we have looked at so far, you should have been able to gather that

a. The wires have to be connected properly within the computer.
b. The proper wires must be on the proper pins in the connector.
c The software driving the port, whether it is in ROM or RAM, must sense for and provide the proper signals in the proper order to satisfy the handshaking requirements.

If any one of these factors is ascew, the things just won't work!

In closing this general section of the discussion, let me say that I had thought about publishing a complete pin breakdown of the RS-232C standard, with functions and origination point of each signal, but I rethought that idea and decided that this information would be too technical for most of the readers who browse through this memo. If anyone would like the breakdown, write to me and I will send it to you. (Enclose an SSAE)

B-128 SPECIFIC INFORMATION

Because of the handshaking requirement mentioned above, it is important to note that any description of how a particular machine drives the RS-232C port in it is totally dependant on how the software is written. Therefore, this description is based on a B-128 loaded with B TERM Software and driving a modem. It is beyond my level of expertise to be able to test to find out if the B TERM software in RAM, or the more permanent software in ROM is responsible for the handshaking sequence that I am about to describe. If you use a different terminal program, and have any difficulty following the guidlines set below, then it is probably because your terminal software is driving the handshaking sequence differently.

Now on to some specifics.

When you power up the B-128 and test for voltages on the DB-25 connector you will find the following;

Pin #2	Transmit Data	Negative 12vdc
Pin #4	Request to Send	Negative 12vdc
Pin #20	Data Terminal Rdy	Negative 12vdc
Pin #24	Sync. Trans. Clock	Negative 12vdc

The only one of these signals that might be a problem is the voltage on Pin #20. Some equipment may require a positive voltage on that pin in order for the connecting piece of equipment to turn on its send/receive lines. If you have any difficulty, this pin is the first place to look and can be changed to a positive signal by making a special cable that, in essence, cheats the piece of equipment that you are connecting to the B-128 into thinking that it is really getting its +12vdc from pin #20. (What you really do is run a connection from pin #9 or #10 on the non B-128 piece of equipment back into pin #20 on the same piece of equipment - questions? If so, write.)

Now, after B TERM has been loaded, when the signals are checked everything stays the same! No changes at all will be detected in the signals, that is until you start providing the B-128 with the handshaking signals it is looking for! For example, before the B will transmit a single bit, it wants to see positive voltages on several pins;

Pin #5	Clear To Send	Positive 12vdc
Pin #6	Data Set Ready	Positive 12vdc
Pin #8	Data Carrier Detect	Positive 12vdc

You must make sure that you provide these voltages from some source. Once you do you will see a marked change in the interaction with the modem or printer you are trying to connect. Now, after you select the terminal mode in the B TERM software, you will type a character and you will see the voltage on Pin #4 (Request to Send) go to a positive 12vdc (it was a negative 12vdc). Immediately after this the computer will check to see if there is a positive voltage on Pin #5. If so, the port will open and spill all of it's juicy bits out into the waiting arms of the peripheral device you have attached to it.

Now, what do you do if the peripheral you are attaching doesn't provide all these signals properly. Weeeeeeeelllllllll, funny that you should ask. Think about it with me - didn't we just finish saying that Pin #4 went to a positive voltage just as we transmitted our first character in terminal mode? What would happen if we just tied Pin #5, Pin #6, and Pin #8 all together with Pin #4? Wouldn't that provide the computer with the positive voltages that it needs just at the right moment? Well, it turns out that it does! A special cable made up to tie 4,5,6,8 and possibly 20 all together will make the B-128 think that it is connected properly and it will start spitting out bits.

I realize that this has all been a bit technical, and believe me, I have left a lot unsaid, but if you try this and it doesn't work, give me a note decribing the problem and I will try to help.

That's it for now. CHOW!

A note from CBUG.

This file is another example of the use of telecom. At the time it was written we were but 7 hours from deadline for inclusion in Pre-Release Disk 1. There was not the time nor opportunity to mail the disk nor was anyone available to make the two hour drive.

Solution is obvious, a two minute phone call between two computers and it was in our hot hands!

As CBUG grows, we will be moving publication files about the country in this manner quite frequently. You too may find it handy.

One of our members is now working on turning the B128 into a remote host computer not just as in a Bulletin Board, but actually emulating a mainframe host with multi-tasking and all. One means of communication between the dumb terminal and the smart host will be thru the RS232 port, and phone lines as needed!

- - - - - - - - - - - - -

Postscript from Ralph Ricks

Thanks to Larry for the help. The Hayes Smartmodem 2400 requires plus 5 to plus 20 volts on pin 20. Both the Commodore Business Machine B-128 and the WANG PC put out minus 12vdc on pin 20 until certain handshaking conditions are met. Then pin 20 goes to the desired positive voltage. There are only a few other wires to worry about with the B-128:

Pin 1: Chassis ground--Nice to have, but redundant if you use grounded receptacles and power cords with grounds.
Pin 2: Data out from computer--Gotta have.
Pin 3: Data in to computer from modem--Gotta have
Pin 4: Request to Send--used by Hayes 2400 only in synchronous mode.
Pin 5: Clear to Send--signal from modem to computer--optional
Pin 6: Data Set Ready--required to be positive before most terminals will begin. The Hayes 2400 holds this line down untilphone is off-hook, etc. This is the biggie for the B-128 with B-TERM. To make the B-128 work with the Hayes 2400, open your cable and cut the wire to pin 6.
Pin 7: Signal Ground: Gotta have
Pin 8: Data Carrier Detect--Signal from modem--gotta have.
Pin 20: Data terminal ready--sometimes it's positive, and sometimes it's negative. This signal comes from the computer.

So what's the minimum number of wires in your RS 232 cable? Just pins 2, 3, 7, 8, and 20.

What's the maximum with the B-128/Hayes 2400 combo? 24--Just nip pin 6 and you're in business. NOTE: not all cables have 25 wires--most modem cables have just pins 1 thru 8 and 20.

It is interesting that B-TERM won't work if pin 6 is held low, but the terminal program that is part of CALC RESULT just ignores that pin.

Comments and questions on the above can also be sent to: Ralph Ricks, 8[illegible] . Sepulveda, Carson, CA 90745

BITS ABOUT BMON

by: Larry Scott

Hi. I'd like to take up some of this valuable newsletter space to clear up a few things about BMON. You see, when I wrote BMON, I did it for myself, but my friends suggested that I should give it to CBUG so that others could use it. I liked the idea that something I wrote might be of use to others, and I liked the idea of getting something in return for the year I spent working on BMON, so I wrote some instructions and put together a disk, which is currently being offered by CBUG. I didn't, however, realize how busy Norm and the others at CBUG headquarters were, so I just assumed that they would print a description of BMON, or at least print the BMON Quick Reference Guide that I put on the disk. So Norm suggested that I write this article.

If that sounds like I'm complaining, I'm not. As a former newsletter editor and a former officer of a user's group, I feel that I am qualified to say that Norm is doing a very good job with the user group and with the Scratch Pad. I know it is a very difficult and time consuming job.

First, about the blurb that told about BMON. It should have mentioned that BMON is written only for the B128, not the CBM256. If there is any interest, I could probably convert it for the latter machine.

BMON has to be in bank 15, but it does NOT require Gary Anderson's cartridge. I can not imagine where that idea came from. As a matter of fact, about a fourth of the BMON users have BMON installed in prom (programmable read only memory) in their Calc Result cartridge, about half have it in prom in some other kind of cartridge, and about a fourth are using it with a ram based cartridge such as Gary's. The advantages of having it in prom are that it can not be accidentally overwritten and that it is available and enabled when the computer is turned on. Gary's cartridge is a very good way to implement BMON, but not the only way.

I thank whoever gave BMON the compliment. "Far advanced professional" may be more than BMON deserves, but it pleases me to no end that somebody thought that highly of it.

Now for a little(?) bit about BMON. It started when I decided to correct the problems that were plaguing Superscript II; the key bounce, the erratic cursor movement, and the disk drive chatter during load. I found the built-in machine language monitor inadequate (to say the least), so I decided to whip up a monitor of my own. A year later, I had BMON. It started with the disassembler and assembler routines from Micromon, and grew every time I sat down to use it and ended up saying, "Gee, I wish I had a command that would ..." Eventually, I had to call a halt and get back to fixing Superscript II, but fortunately Liz Deal had just published her fix. With two problems down, the third didn't take long.

Just what is BMON? BMON is a monitor program. It allows you to look at and alter the contents of memory. BMON is designed to aid in converting the many programs written for other 6502 based machine, like the Commodore PET, CBM, 64, or Vic-20, to run on the B128. It has some features which will make it usefull in debugging any machine language programs you may write, but it is directed primarily towards the conversion process. If you don't write your own programs, or if you deal only in basic, BMON will not be of much help to you.

BMON has commands to save files, load at an optional address, report the load address, end address, and length (in bytes), and verify that a file is an exact copy of memory. BMON also has a command that lists only the file names in a directory, to allow you to cursor up and load with one keystroke. Also, the B128 built in monitor's @ command is still enabled.

There are also commands to allow you to disassemble, dump, fill, alter, move, and compare memory, even across banks. You can hunt through memory for any combination of hex bytes and ASCII text, any machine code that references any area of memory, or any illegal opcode. These last three commands facilitate converting programs to run on the B128. One command allows you to quickly find all data tables, and another lets you find all references to (for instance) the i/o chips.

I have built into BMON a hexadecimal/decimal/binary/ASCII calculator that evaluates expressions the same way most assemblers do. It will add, subtract, rotate left and right, and do the logical operations AND, OR, and EXCLUSIVE OR. Also, the result of the previous calculation can be used in the present calculation by using a special character as one of the numbers.

BMON supports a printer at IEEE bus address 4. There is a command that allows you to output any combination of hex bytes and/or text to the printer. For instance, one command line could turn on the printer page mode, set enhanced mode, output a title, and do two carriage returns. Any of BMON's commands can direct their results to the printer.

Several of the otherwise unused control keys are utilized. You can scroll up or down through a memory dump or disassembly. You can disable, exit, or kill BMON. Or you can dump the screen to the printer(even when in BASIC, as long as BMON is enabled).

Two commands are my personal favorites. One is a complete cold start. I like that one. The other one is for when BMON can't carry out a command, whether it's because of a syntax problem or because of a bad memory location (like if I accidentally try to move a program into bank 11). When that happens, BMON puts a question mark at the place in the command line where it ran into trouble. If the problem is not obvious, you can use the WHY command, which tells you, with text, why BMON could not do what it was told.

To specify an address in BMON, you can use from one to six digits. If the address is within the current bank, the bank number is optional, as are leading zeros. A five or six digit number specifies the bank as well as the address. There is a command for setting or finding out which bank is the current one. All addresses listed by BMON include the bank.

The BMON calculator will not work with a CBM256 because of the differences in the BASIC roms. If there is any interest, I can fix that quite easily.

The BMON commands were written to use the screen windows.

Enough of BMON. <<and on to some more complexication of the 8050 diagnosis>>

Concerning the 8050 disk drive reliability: there is a certain integrated circuit on the main logic board that tends to be bad. It is a thing called a cmos analog switch, type MC14066 or CD4066, and is designated UN4 on the schematic. It is very sensitive to static electricity. It is the nature of the part that when it is damaged, the drive will probably still run, but it will have a harder time reading disks. It may become temperature sensitive, that is, it may work when you first turn it on, but fail when it warms up for a few minutes. Conversely, it may not operate 100% until it has been on a while. Since the DOS compensates for flaky disks by re-reading bad tracks, the problem may be present for a long time before the drive starts reporting any errors. When you access a file, watch the error/power led. If it ever flashes red, you have a problem, and it could be this one. The part is usually in a socket, and is easily replaced. If it's not, a socket should be installed when the part is replaced. Once you get a good one, you will probably not have any more trouble, but Commodore seems to have gotten hold of a batch of marginal parts. That happens sometimes, and usually doesn't show up until the parts start going bad months later.

Unfortunately, this problem has the same symptoms as a speed problem. Also, re-adjusting the speed may hide this problem, since it will increase the odds of getting a good read from the disk. However, the 8050 drive is not usually touchy about speed, and the disk speed has to be quite far off before it becomes noticeable. If a drive appears to be overly sensitive to speed variations, and the problem is not the zener diode that Warren Kernaghan has been finding, it may be this bad analog switch that is at fault.

<<The speed sensitivity of 8050 drives appears to vary radically, drive to drive. Some are quite usable and have run for 4 or 5 years without a hitch, others drive you up the wall every few hours, if not from the moment of turn on. The above info from Gary will add to our rapidly growing knowledge of how to keep our systems running.>>

MODIFYING SID's AUDIO AMPLIFIER FOR BETTER SOUND

by: E. Todd Taylor

When I first bought my B-128 from Protecto and discovered that it had a digital signal processor to sound device, better known as SID, I was delighted. A whole range of possibilities were available such as security and frequency controlled manipulations of devices. If only I could hear it!

I performed a continuity test on the amplifier circuitry as well as a diode test of the Darlington pair transistors: all was in order. This led me to believe that either the maximum power transfer between matching impedances was incorrect or, hopefully, that the gain was too low. I guessed that the gain was too low. So, I looked around for an audio amplifier circuit in some data catalogues and found one.

Using National Semiconductor's LM386 and its corresponding "typical application" circuitry, I replaced the existing B-128 audio gain circuitry.

First, I built the new amplifier on a small piece of perforated proto-board, all except the connections into the B-128 circuitry.

Next I had to decide where to put my new amplifier. I cut R38 from the circuit (ie. cut the lead from the R38-R39 junction) and I added the input from my circuit, as well as cutting cathode (-) of C30 from the circuit and adding the output from my circuit: the new amplifier was in place.

Finally, supplying 12V and ground came from anywhere I could easily tap it. For 12 Volts, I soldered to the collector of Q5 (or use Q6) and stole a ground reference from the ground rail which surrounds the system circuitry.

I couldn't just leave the new amplifier unprotected, so wrapping it in a cardboard shell or some other non-conductive material (don't use tape, since it leaves a gummy residue) and placing it on top of the P6 and P7 connectors provided a relatively "clean" (but ugly!) way of securing the amplifier.

```
 10 si=55808:fl=si:fh=si+1:w=si+4:a=si+5:h=si+6:l=si+24
 20 bank 15
 30 poke l,15
 40 poke a,9
 50 read x:read y
 60 if y=-1then poke w,0:end
 70 poke fh,x:poke fl,y
 80 poke w,17
 90 for t= 1 to 100:next
 100 poke w,0
 110 for t=1 to 10:next
 120 goto 40
 130 data 8,146,9,159,10,205,11,113,12,216,14,106,16,46,17,37
 140 data -1,-1
```

which comes from the B128 Programmer's Reference Guide (p.40). If it works, one should hear a quick increase in pitch.

Change line 60 to:

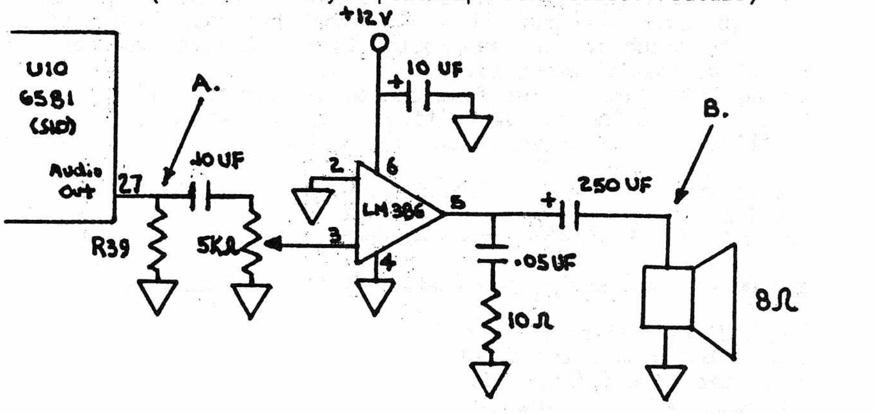
```
60 if y = -1 then poke w,0:restore:goto 10
```

so that adjustments can be made to the 5K ohm potentiometer to an appropriate level of maximum volume.

Now that I can hear what I am doing, I have since modified a few programs meant to run on a C-64, I have created a small library of sounds and basically discovered the world of SID.

I am willing to sell the finished circuit and connections for a moderate fee (which includes shipping and receiving, parts and labour), and all that has to be done is to solder 4 connections and cut 2 components on the system board as part of the final installment.

For more details, write (or phone): E. Todd Taylor, Box 434 Russell, Ontario K0A 3B0 CANADA Tel:(613) 445-5413
Note: I also sell memory upgrade kits to 256k (addressable by superscript bank select feature)



A. R38-R39 JUNTION (NEW I/P)

B. C30 (-) (NEW O/P)

Replaces original gain cct. of SID audio out.

HINTS & TIPS

MULTIPLE COLUMN PRINTING: Q. from member Fred Peterson regarding Superscript multiple column set up. When ESC O C S is used to produce a "background" print file in building a narrow column text, the file comes back as garbage.

A. ESC O C S absolutely must be used ONLY in the default printer mode (CBM) else the file saved will be coded for some other printer -- non english on the screen when brought back for further editing.

You can "restart" Superscript by using ESC SHIFT CLR, Y, Y. Then restart the program with your current printer selection. The text then on screen will be preserved if you do not change screen width.

8050 PERFORMANCE DISK: The 8050 demo disk that came with most drives, and is reproduced on Norm's Utilities, CBUG #1, the program "performance" which does an analysis of the 8050's performance DOES NOT check speed alignment. If the drive is so far off speed that it is unable to properly new, write or read a disk, the performance program reports "read error" and instructs you to try another disk. That instruction is not necessarily to be believed, as 99% of the disk problems reported to CBUG are actually drive speed problems. You MUST use Physical Exam or equivalent to properly test for speed and head alignment. The "performance" program is an inadequate attempt at a go/no-go test; but is valuable as it does a full read write to an entire disk.

ORDER ENTRY/INVENTORY: Member Bruce Faierson of Northwest Music & Computer (also an advertiser) will soon release a complete inventory, order entry and receivables package which operates within Superbase. The program will resolve the point of sale and inventory number problems some members have experienced in the Info-Designs/Cabs programs. Anyone else developing such applications programs are urged to make them available thru CBUG.

INFO-DESIGNS/CABS BACKUP Many members using this accounting suite have failed to figure out how to make backup copies of their program disks. Yes, the disk and book say it can't be done, but that is not the case with the B128/8050 combination. First though, a point of good practice, and absolutely necessary with some of the CABS disks: apply a write protect tab to the original disk BEFORE doing the backup -- some versions had "worms" in them which would destroy the original if a copy were attempted without the tab!

When you attempt to make a copy of a program disk using the backup provision within CABS, the B128 refuses to do so and tells you this is illegal! BUT all that is happening is that the CABS backup program looks at the disk header name and refuses to copy one that is a CABS program disk. You can do backup from any other function. NOTE -- backup is different than COPY. Backup replicates all the data on the disk including the random files which CABS uses, whereas COPY not only does not copy random files, but jambs up if you attempt it.

As a matter of personal preference I do most of my routine backup and copy work from Superscript's disk mode simply because at the end of the procedure, there is an on screen report of the status of the copy -- OK or bad, and if bad why! In Superscript enter disk mode SHIFT OFF; put your original (with tab installed) in drive 0, and a blank disk in drive 1; then type d1=0 RETURN . You then answer "y" for yes to the prompt and off you go. If you get a speed error report, put the original in drive 1, the blank in drive 0 and enter the command d0=1 RETURN .

Working copies of the CABS program disk can not have a write tab installed because data is written to and from the disk. Obviously to risk your only program disk is worse than foolish. We strongly suggest you take your original set of programs, make duplicates of them. We then call the original purchased copies the Vault Masters, and never ever (hopefully) have to touch them again. Wrap them in a zip lock bag and put in a safe, secure, and temperate place. Now take the copies, put tabs on them, and label them Master Copies, and make a set of copies from them. This last set, 3rd generation, is the working copy. If you ruin one, you can go back to the Master. And if, heaven forbid, you ruin the Master, you can go to the Vault copy.

IF AT ANY TIME YOU SUSPECT YOUR DRIVE IS MESSING UP DISKS, DO NOT RISK YOUR BACK-UPS. CALL FOR HELP AND GUIDANCE BEFORE PROCEEDING!!!

SFD1001 For the few of you using this drive, it will load most B128 programs, however you have to attempt loading twice so the drive can "learn" that you have installed a single sided disk. People with experience with the SFD are invited to write in with tips on using and caring for them.

SERVICE MANUALS: Commodore has provided CBUG with an extensive list of SERVICE MANUALS. Below are those of interest to B128 Users. Each is $25.00 plus $3.00 shipping per order. Prices and availabilities subject to change!!! I suggest you make each manual a separate order and check per chance one of several is out of stock.

Part #	Product	Part #	Product	Part #	Product	ORDER FROM:
314003-03	1526/802/4023	314011-03	8050/8250	314012-04	6400	CBM Parts Department
314010-06	B Series (B128)	314011-04	9060/9090	314012-05	8023	1200 Wilson Drive
314011-02	2040/4040	314012-02	4022	314012-06	8300	West Chester, Pa. 19380

GROUNDING & CABLES: As people expand their systems, or undertake surgery on their drives certain information is important. An operating system MUST be mutually grounded via the power cords. If all units are plugged into a power strip, then they are mutually grounded. Do not rely on wall outlets. Earth grounding via the power cord is usually adequate and not critically important in most installations except as to safety. If you have a high static problem such as low humidity and wool carpets, you MUST address the problem by providing proper earth grounding and SEPARATE means of discharging yourself to ground -- example: install aluminum burglar alarm tape around the perimeter of your equipment and always discharge yourself to this tape before touching your computer. Of course the tape must be DIRECTLY GROUNDED, not via the computer ground.

That aside, on to cables, cartridge and the like. The cartridge port provides power to the chips in the cartridge. As such, if you insert or remove cartridges while power is turned on, you will likely destroy the cartridge, and possibly hurt the computer. This is because of the high risk that the ground pin will disconnect before the power pin. NEVER insert or remove things in the cartridge slot while the power is on!

The RS232 port also has several power pins available. Though normal modem use only uses the two ground pins and pins 2&3 for signal, some other devices DO use the power take-off pins. If your device is using the power pins, you MUST make sure to turn the power off before opening the cable. Ditto for the Cassette port which is currently used for the "dongle" (protection cartridge/key) in the CMS accounting system.

NOW, the GOOD news. The IEEE-488 buss has NO power in it. You can unplug it at any location at any time signal is not being transmitted. That means the drive and printer are not in use. Take care to properly ground yourself before handling the open cables, lest you cause a static discharge to the open lines. This capability is very important if you are going to do surgery on your drive, as you should load Physical Exam into the computer before starting, leave the computer on while you disconnect and repair the drive. Upon completion you merely have to turn on the drive and plug in the IEEE cable and go!

The IEEE buss is a party line system. It will tolerate many devices on line at the same time so long as each one turned on has a different device number and no more than two devices on line are shut off. This two device rule does not apply to some older printers however, and applies to interface devices such as IEEE to Centronics converters only if they are powered from the printer. If the buss is loaded by too many unpowered devices massive errors will occur -- you'll know it at once.

THE PET CONNECTOR TO THE B128: Many members have complained that the IEEE plug inserted in the back of the B128 falls off or is loose. Some proudly told me of glueing in place. DON'T, DON'T, DON'T!!!! Never glue as the adhesive can get into the plug and act as an insulator between the contacts. Rather use filament (shipping) tape and strap the connector firmly in place. Of the connectors in the IEEE cables, the PET end (an edge connector) is by far the most fragile.

GRAPHICS IN SUPERSCRIPT: Many members have asked how to underline, etc in Superscript. On a 4023 or 8023 you can use CONTROL 3 or 4 to create underlines or overlines. About a dozen of the graphic symbols on the keyboard will reproduce on the printer. Underlining of text is a bigger headache as it has to be done from an additional line, seemingly restricting you to double space. To set up text for addition of underlining you should complete the text normally, the ESC O C S to create a fully formated file, double spaced of course (*sp1) . Then load the resulting file back into edit mode and insert the underlines using the overline characters from the blank lines. We are expecting an extensive article on this subject persuant to development work done by Lou Lohman and others on our DELPHI network.

SUPERBASE TIPS: From Charles Tyson: The "SELECT DELETE" command which works in the C-64 version does NOT work in the B128 version. You must issue the command "SELECT D" to delete a record from within a program. Even if deleted, disk space is not freed up -- the space can only be reclaimed by Exporting and Re-Importing into an empty or clean database.

From Dennis Kreuzenstein: Superbase will display number with 4 decimal places but you need to tell it to use the "&x,y" format found on page T-39 then the variable you want to print or display. Example: A=1234.5678, command Display &4,4;A would return the number 1234.5678

Again, we invite members to send in tips, hints, bugs whatever on our software.

Tips from our librarian, Mark Schwarzbauer

PRACTICE WHAT I PREACH!

TIP 1

WHEN DOING A HEADER COMMAND- if you get the error reading "bad disk" don't believe your B. Most likely this is a weird indication of a format speed problem- honest! Check format speed and correct if necessary before throwing out the disk.

TIP 2

Areas where power fluctuations may occur frequently, may produce format speed problems. I can't back up this theory, but for the sake of your equipment, if you are experiencing this regularly be sure you have a good surge suppressor on your equipment. This should go without saying.

TIP 3

For format speed adjustment, you can follow the procedures in my instructions, and to speed up the work, you can try to adjust the speed without removing the drives. To do so, you can try to read the white hub wheels that lock the disk down. On most, if not all versions, of the 8050 these are also marked, although the markings are more difficult to read. I recommend you follow the complete procedure first and then use this faster method for later work. I no longer remove my drives and can adjust my format speed in aoout 1 minutes time!

Several hints from member Robert Houk (Jasper & the PAK)

When programming and having to break into the program at an input, hold run/stop and then return. This is obvious to those who have used it, but to those who reset every time, it is a lifesaver (finger life that is)

Trace the Superscript function Key card and cut out several, then label them for your own set of programming and the Modem presets you make <<See want ads "For Sale" for cards>>.

Dsave@ is also obvious, but who reads the book.

To alphabetise, (FAST), and list, just type the words into Superscript and let the spell check do it in no time to the printer.

From member Jon Whatley: and improvement to Joe Rotello's drive speed up, add lines 1, 2, 3 as follows:

```
        1 open 15,8,15
        2 print#15,"m-w"chr$(0)chr$(16)chr$(3)chr$(6)chr$(4)chr$(250)
        3 clr
```

I have also enhanced line 200 to reset the printer case mode to normal when signing off the Menu with "E". This allows immediate loading of SB or SS so that it will print in upper/lower case:

```
        200 ifa$="e"thenopen10,4,10:print#10:close10:end
```

One may also want to add a print clr/home to clear the screen in lines 3 and 200. Inserting this at the top of Liz's Pre Superscript at line 300 and then a line 305 clr insures the printer is properly reset.

From member Anthony Goceliak, Jr.

It is always my goal when writing programs to make their use as obvious as possible to a user, possibly a holdover from the time before computers when I built a radio transmitter with some 20 odd controls on the front panel (none of which were labeled of course). It worked (actually it still works) just fine, but after I had gone on vacation and re-confronted all those unlabeled controls ...?!... well, I guess you get my point....

As to Tony's tribulations with his 8050: ... I fixed the drive in the meantime. The problem was the 6432a I.C. UE1. As many people out there already know, this chip is manufactured by Commodore's on chip subsidiary, but I found a source for somewhat lower rated "not quite" replacements. The work somewhat flakily in the disk drive, but work just great in a 4023 printer. Swaping the chips is double work, but at the end you wind up with a working disk drive and a working printer.

From member Steve Stone: The RS232 port on the B128 is missing the connection on pin 18, which is known as the Ring Indicator. <<This is of interest only for automatic answer, and is more easily handled (I'm told) by software interface with the modem.>>

From member Edwin Bowerman: Ed had experienced problems in Superscript trying to use the *ch command to create user defined characters such as subscripts and degree signs. From Brian Leighfield of Precision, Ed was instructed that to avoid the problem with Superscript (uses a 8x6 matrix but the 4023 using an 8x8 matrix), use *sa5 instead of *ch. "Defining a degree sign as *sa5,96,144,144,96,0,0,0,0, and ESC $ to enter the user defined character, we can write with a proper degree sign. Defining a "delta" (Triangle) is *sa5,2,10,34,130,34,10,2,0. The trick of putting two user defined characters on a line requires using the *ps instruction to stop the printer at the line in question, manually backing up the printer paper (nine clicks of the detent on a 4023), defining the second character and restarting printing with "p".

USING INPUT# FROM THE SCREEN

by: Art Klinger, 25 Jan 1986

To get around some of the disadvantages of the normal INPUT, users of any of the established Commodore machines can treat the screen as a IEEE-488 device, by OPENing a file channel (fn) to screen device 3, then using INPUT#fn and PRINT#fn, t gather or print information to the screen. This article explains some of advanatages and differences of the two methods, demonstrates some B128 INPUT# bugs, and develops a short but useful INPUT# routine that provides programmers with some important advantages.

Let's say that we want to INPUT data from the user at the bottom (25th) line of the screen, then place that data at strategic locations on the screen. In other words, the screen is formatted with comments, prompts, highlights and borders which must not be disturbed except as explicitely directed by the program. A quick first attempt using the normal INPUT command might be:

```
                100 print"(home)xxxxxxxxxxxxxxxx";:    rem"cursor to & print at top left scrn
                110 fori=1to24:print"(cd)";:next:      rem"cursor down to screen bottom
                120 :
                130 INPUT"enter the next name";n$:     rem"have user enter the data
                140 print n$;:                         rem"print name just to the right of n$
                150 print"(home)";:                    rem"send cursor home, ready for print
                160:print tab(3)n$;spc(3)n$,n$:        rem"print name 3 times across top line
```

When this is run, we immediately get into trouble. INPUT in line 130 prompts with a question mark even though the prompting phrase is not a question. If the operator accidently puts a comma or other control character into the name n$, the computer scrolls the screen upwards and prints error messages when the erroneous string is entered. Even without erroneous input, the computer forces a cursor return and line feed the moment that RETURN is pushed, which still scrolls our carefully formatted screen. We need an input that doesn't generate the question mark, error messages or line feed.

All three of these behaviours can be avoided by opening the screen as if it were a file, and using INPUT#. A first attempt with this method may yield something like the following, which is line-for-line comparable to the above routine:

```
100 (as above)
110 (as above)
120 open1,3:                         rem"open file #1 to screen device 3
130 input#1,"enter next name";n$:    rem"have user enter the data
```

Unfortunately we neeo go no further than that, since the B128 never gets past line 130. As soon as the data is entered, the B128 jumps out of BASIC with error message "STRING TOO LONG ERROR IN 130", apparently due to a BUG in the B128 INPUT# routine.

So let's command (CMD) the B128 to keep the file open, and see if that makes a difference:

```
100 print"(home)xxxxxxxxxxxxxxxxx";: rem"cursor to & print at top left scrn
110 fori=1to24:print(cd);:next:      rem"cursor down to bottom (25th) line
120 open1,3:cmd1:                    rem"open & command file#1 to device3
130 input"enter next name";n$:       rem"have user enter the data
140 print n$;:                       rem"print the name just to right of n$
150 print"(home)";:                  rem"send cursor home, ready to print
160 print tab(3)n$;spc(3)n$,n$:      rem"print name 3 times on top line
170 print#1:close1:                  rem"close file#1, & back to normal
```

Oddly enough the program does not crash now that the machine is COMMANDed to keep the file to the screen open. However, working diligently with it reveals still more problems, as well as our sought-for advantages. Unfortunately, the screen scrolls even before the INPUT statement, because of a line-feed performed by the CMD command in line 120. Happily, the new INPUT of line 130 now gives a flashing cursor with no question mark prompt, does not print error messages for many of the normal errors by users, and does not perform a cursor return/line feed when RETURN is pressed after the input. However, the INPUT cursor is two places further left than the position taken by all previous Commodore machines (PET,CBM,VIC,C64) for INPUT# or INPUT. This is most noticable when a programmer tries to construct the normal "default" input using the three "cursor lefts" in the form; INPUT"enter yes or no y(cl)(cl)(cl)";a$. With this commanded screen input, we must use only one (1) cursor left, to get the cursor to the proper position. We also see a cursor position problem in the next line, 140, which prints n$ too far to the left, writing over the last letter of the n$ reply just entered by the operator. To fix this, we would have to add a couple of cursor rights just before PRINTing n$. Finally, careful inspection shows that the TAB, SPC and COMMA (,) of line 160 ERASE EVERYTHING IN THEIR PATH, instead of passing invisibly over any existing characters on the screen, a significant bug indeed!

The following routine avoids all of these problems, and at last gives us a useful routine:

```
100 print"(home)xxxxxxxxxxxxxxxx";: rem"cursor to &print at top left scrn
110 fori=1to24:print"(cd)";:next:   rem"cursor down to bottom screen line
120 open1,3:cmd1,;:                 rem"open file1 to screen3,cmd with ','
130 input"enter next name";n$:      rem"have user enter data
140 print"(cr)(cr)";n$;:            rem"move right then print name
150 print"(home)";:                 rem"send cursor home
160 print"(cr)(cr)(cr)n$---etc      rem"use cursor moves to avoid erasures
170 print#1,;:close1:               rem"close file, w/; to freeze cursor
```

Now this functions satisfactorily in all respects. Notice that CMD and PRINT# cuases a cursor return and line feed (and screen scroll if performed while the cursor is on the bottom line), just like the normal PRINT command, unless followed by a semicolon. This should not be a surprise however, if it is remembered that CMD and PRINT# ARE indeed PRINT operations, using most of the same ROM routines. CMD4,"disk title for directory"; is exactly the same as PRINT#4,"disk title for directory"; except that CMD does not close the file number 4 when the the statement is printed. Just remember that the comma and semicolon are both needed to suppress the cursor return/line feed without syntax error. Printed cursor control characters are used to properly position the cursor during and after INPUT, and cursor control characters must also substitute for TAB, SPC and Comma (,) when repositioning while under a commanded screen input.

A user can still break out of the program and mess up a screen if he uses cursor control keys while the machine calls for INPUT. To avoid this, a programmer can place a quotation mark (") under the cursor prompt for INPUT. All cursor control keys are then interpreted as literals and are not acted upon, which keeps the user from cursoring to other locations on the screen. The quotation mark may be made visible or invisible as desired by the programmer. Since most computer users are not used to seeing a quotation mark during input, it is probably best to hide the mark. The final routine includes this feature with the best of the above features. Now that the TAB,SPC bug has been demonstrated, we can also drop the somewhat arbitrary cursor positioning program lines. The resulting input routine becomes:

```
100 rem"*********** improved, bottom-line input routine *************
110 down$="":fori=1to24:down$=down$+"(cursor down)":next
120 down$="(home)"+down$
130 :
140 print down$;:                   rem"position cursor on last line
150 open1,3:cmd1,;:                 rem"open & cmd with (;) to freeze
160 print chr$(34);chr$(20);:       rem"print a quote & delete at input
170 input "enter 2 strings:  ";x$,y$: rem"get the data from the operator
180 print#1,;:close1:               rem"print with (;), close scrn file
190 print"(home)";:                 rem"or otherwise get off of last line
200 :                               rem"then use x,y when and how desired
```

Notice that lines 110 and 120 merely builds a cursor positioning string for easier and faster repeated positionings to the bottom screenline. Line 190 is likewise not mandatory, but gets the cursor off of the bottom line to minimize the chance of a careless program line later causing screen scrolling. Multiple variables can also be input, as shown in line 170. Notice that there are two spaces in the input prompt string in line 170, to get the cursor to a more traditional position for input. The operator can press cursor control keys, reverse key, and delimiters, without causing screen scrolling or program exits. Even quotation marks are calmly accepted into the input x$ and y$. However, at each odd-numbered count of quotation mark key entries, the user is no longer protected against presses of cursor control keys etc. In other words, in the string 'aaaa"bbbb"cccc', the user is protected while entering the 'aaaa' and the 'cccc', but not the 'bbbb'. Also, for some reason, if a odd number of quotes is left in the input string, the cursor will not go "home" as directed in line 190 (another bug?).

The core of the routine is contained in lines 150 through 180, which could fit on just one 80 column line by deleting the REM statements.

In summary, we found at least three apparent bugs in the B128 INPUT# as it appliess to screen file input. A response to INPUT# causes a crash with "string to long error" unless the channel is commanded (CMD) to remain open with the screen being the default device. While in this mode, the cursor position for INPUT seems to be offset two characters to the right of its traditional position, and the cursor position for PRINT seems to be offset two characters to the left of its traditional position. While in this mode, the character used to move the cursor for TAB, SPC and Comma (,) seems to be a SPACE or SHIFTED SPACE or other character which erases all in its path, instead of skipping harmlessly over existing screen characters. The best solution to these bugs seems to be to simply avoid TAB, SPC, Comma or PRINT while in this alternate mode, and to get out of the mode as soon as possible. With these precautions, the method does its job well. The input location is shown by a flashing cursor, but has no question mark. An error such as forgetting following numbers for a multiple-variable input does not result in error messages or screen scrolling. There is no cursor movement or line feed or screen scrolling when the reply is entered, which allows INPUTs from the last screen line. Finally, the inclusion of a hidden quotation mark makes it far more difficult for operators to cursor off-line or scrol the screen when they accidently press cursor control keys. In spite of the several B128 input bugs, the technique provides a very valuable routine, and can be used on most if not all Commodore machines made before 1986.

Art Klinger, 4659 Balboa Dr., Wichita Falls, Tx. 76310

HINTS & THOUGHTS

by: Art Kinger

Norman, here is collection of miscellaneous hints, kinks and thought-provokers that may be of interest to your readers. In those cases where an unsolved problem is indicated, possibly some of your readers would have solutions. Many of these bits may be better suited for airing in your newsletter rather than just retention on a disk.

KERNAGHAN'S UTILITY deserves to be a model for all of us! His combination of short and long instructions, his organization and labeling of program types in the disk directory, his thoughtful selection of programs, and his thorough adaptations and debugging, indicate a public-spirited professional. Many people won't appreciate the great amount of work put into that disk, but because of that work, his collection will see wide use and live a long time. Congratulations and much thanks, Warren!

FREEWARE AND ROYALTY REIMBURSEMENT: Norman, please clarify again. In which cases (which disks and programs) do we owe these fine writers/programmers some dues above and beyond any extra charges that you might have charged for a disk. I sure want to make sure that Liz, Warren and other workers get at least these token payments for their unselfish work.

<<I'll address this subject in the library lead page.>>

B128 SERIOUS APPLICATIONS: As Norman pointed out, CBUG's members seem to be an enthusiastic bunch dedicated to getting the most out of the B128, and putting it to serious applications. As an avid engineer, I am particularly interested in the possibilities of engineering, statistics and graphing programs, and any methods of real-time monitoring and controlling of machines or processes via analog and digital input/output. I hope that others interested in such subjects will communicate through CBUG or drop a line to me, Art Klinger, 4659 Balboa Drive, Wichita Falls, TX 76310.

ROUNDOFF ERROR ELIMINATION: A computer's roundoff error can and often does introduce significant error into calculations, especially when they are repetitive (reiterative). Not well known is the fact that roundoff error can be corrected easily in apparently all machines that use Microsoft BASIC, by converting the number to a string and then taking the VALue of that number. The following routine demonstrates roundoff error and its correction. Now if we can just find a mathematician/programmer to EXPLAIN AND SHOW IN DETAIL, just what this trick can do for us, and how to use it in our statistical, regression and other numerical programs. Is it possible to do perfect math - reiterative computations with absolutely no accumulations of error? Lines 100 and 120 of the following routine show the computer accuracy with and without the correction:

```
100 for i=0to6step.1:printi;;:next
110 print:print:print
120 for i=0to6step.1:i$=str$(i):i=val(i$):printi;:next
```

DISK DRIVE PROBLEMS: Page one of the Jan 22, 1986 CBUG Scratchpad comments on apparent alignment problems with the 8050. In many if not most cases, it may be unfair to accuse the 8050 of alignment and temperature-drift problems. Since the introduction of the PET with 4040 drives in the late 70's, and the CBM8032's with 8050 drives a little later, there have been complaints about read errors, alignment problems, and drives that mysteriously damage themselves whenever they are moved. Hidden in the multitude of complaints were a few voices that mentioned that such problems could almost invariably be corrected by merely sending the reading head back "home" by initializing or formatting a disk, as suggested by CBUG. This 8050 user only had such problems when the drive was moved or jostled, or when it was turned off while running or otherwise oddly manipulated by my teen-age son. Finally I learned to reset the head (send it back to its "home" track position) whenever the drive miss-behaved, and I have never had problems since. The performance of my venerable 5 year old 8050 has been unbelievably reliable through thousands of programs copied or saved to hundreds of disks, in a room that varies widely in temperature.. My disks are almost all inexpensive SINGLE-density disks costing less than $1.35 each. I have NEVER had read or write errors that were not traceable to either this problem or some other error that I myself created. My drive is the older MICROPOLIS 8050 with DOS 2.5, but I would wager the head homing behavior and reliability of the newer drives are comparable. So if you have read/write problems, first check to see if you have too many peripherals connected with cables that are too lengthy or poorly connected. If the problem persists, try the following:

1. Make sure there are NO disks in either drive.
2. If the problem is in drive 0, enter:
   headerd0,"xx",ixx
3. If the problem is in drive 1, enter:
   headerd1,"xx",ixx
4. Wait about 1 minute while the drive initializes itself and tries to format the nonexistant disk.
5. To turn off the resultant error light, enter:
   print ds$ (RETURN)

4040 DISK DRIVE PROBLEMS: Contrary to the 8050, some of the early 4040 dual disk drives and older 2040 disk drives had read/write problems allegedly caused by an imperfect spindle or bushing - the hub that fits into the disk's center hole to center and hold the disk as it is pushed down on the read/write head. These were the early Shugart model SA390 drives. I need a couple of these easily installed replacement hubs, but my only known source (in Dallas) refuses to sell me one, i sisting instead that I crate up the entire drive, send it to him, and pay $50 in addition to the transportation costs and time losses, for him to replace the one bad hub. Can anyone tell us 4040 users a little more about this problem, and where we might get good hubs economically?

THE VERSATILE 1571 DISK DRIVE FOR B128 USERS: The 1571 handles three disk formats, the 1541 for C64 users, the higher density dual-sided for C128 mode, and an industry standard format for CP/M mode. Although it transfers information serially, its speed in the two advanced modes is at least as fast as our parallel IEEE-488 bussed 8050's. The B128, CBM8032 and PETs have Serial-to-Parallel and Parallel-to-Serial converters built in, in the form of the tape input/ouput system. Also, I believe the hardware/software systems of both the C128 disk buss and the PET/CBM/B128 are often referred to as "Commodore's IEEE-488 based Serial input/output", implying similarities in their implimentation. With proper machine language routines, possibly even with just POKEs to change speeds and mark/space ratios, would it not seem possible for us to simulate the C128 disk buss through the PET and B128 tape ports? Alternately, we could build our own converter. Inexpensive integrated circuits are available to transfer parrellel bytes to serial and vice-versa. One or two extra chips might be necessary to handle the "attention" lines or recognize the addresses called by our IEEE buss , or if necessary things could be switched mechanically/manually by the operator. Access to the inexpensive 1571 single and 1572 dual disk drives would open up a world of CP/M, C64 and C128 software for B128 users. Anyone able and willing to show us how to impliment this, preferrably through the existing serial tape port?

COMMODORE PRINTER REVIEWS, LACK OF: I subscribe to or purchase almost every magazine that even occasionally has a word about Commodore equipment, and a few that do not. Yet in the past 3 or 4 years I don't think that I've seen one word about ANY of the number of different Commodore printers and plotters that have been offered during this period! Retailers also seem to avoid them like the plague, instead offering almost any other feasable substitute that may or may not have comparable price/performance ratio. Yet my experience and the little information that I have indicates that most of their printers and plotters have been worth considering, especially for Commodore user's that don't like the hassle of extra interface cables and adapters that don't always properly emulate all of the Commodore printer functions. Is there a world-wide conspiracy of silence on this subject? Does Commodore not care if their printers are publicized or sold? Can anyone list all Commodore printers and plotters (obsolete as well as those still being sold), along with their present price/value, the name of the Original Equipment Manufacturer (OEM) and the closest corresponding OEM model number, their features, strengths and weaknesses, and where they might still be purchased? I am particularly interested in their newest and best dot matrix and letter quality printers, such as the CBM MPS1000. It sounds like some may have features amd versatile interfaces to fit any and all Commodore computers. Enough of the secrecy now! What's going on?

BUG?- POOR SCREEN UTILIZATION: My B128 is displaying this output on a "13-inch" XTRON AG12X greenscreen purchased from Protecto. It is a high resolution screen (1000 lines center, 800 at edges) that has a built-on tilt-and-swivel base and an attractive, modern design that matches almost perfectly, the color and styling of the B128. However, the combination ends up putting all of the display in an area of only 7.2 inch wide by 5.9 inch high in the center of the 10W by 7.5H total screen size. In terms of area, this means that 42.5 square inches of the 75 square inch screen is being used - i.e., a waste of 43 percent of the available screen! The result is that the displayed letters are crisp and clear, but a bit small and scrunched, especially in the horizontal direction. Protecto and Xtron people seemed interested but unable to suggest a solution (except possibly via an exchange), and the theory was that the problem was mostly in the B128.

Notice that the B128 display has no space between lines of text on the screen. In other words, the decenders (y,g,q, etc) on one line reach down and touch the ascenders (P,l,t, etc) in the line below. In other words, there is no blank raster line separating the character lines in either the normal Text mode or the Grphics mode. The lack of separation is fine in Graphics mode when it is desireable to connect the characters into lines and shapes, but it makes for a more difficult or tiresome screen when used for normal Text. User's of the old PETs and CBMs know that those fine screens have a blank raster-line (pixel) space between each line in text mode. These older machines manipulate the Video Control chip (VIC chip) to automatically omit the spaces and compact the screen when the computer is switched to Graphics mode.

Users of the B128 can gain a more readable display and use more of the available screer by poking register 9 of the B128's 6845 Video Control (VIC) chip to add more spaces. Poke 55296,9 to get at the vertical spacing register, and then Poke 55297,8 to get the proper one pixel spacing for text. Experiment also with poke Poke,6, 7, and 9. A poke of 6 gets some overlapping of characters, 7 is the machine's normal default value, and 9 puts 2 spaces between lines, which will probably spread the display enough to lose some lines off of the top and bottom of the screen. The screen's "vertical hold" may have to be re-adjusted slightly to hold the spread screens steady. When someone provides us with a "ROM upgrade", I'd like the machine to automatically switch the VIC register from 7 to 8 when the mode is switched from Graphics to Normal Text (via the keyboard mode key or a printed chr$). This would cure the crunch and screen waste in the vertical direction, but we still desperately need a hardware or software fix for the severe crunch in the horizontal direction.

BUG - INPUT# FROM SCREEN: To get around some of the disadvantages of the normal INPUT command, Commodore user's can treat the screen as a device, and OPEN a fle channel to the screen (device 3), then use INPUT#fn and print#fn, where fn=the file number. For example, the following routine does not print a question mark for the prompt, does not print error messages for wrong inputs, and does not perform a carraige return/line feed on the cursor when the response is entered:

```
100 open1,3
110 input#1,a$
120 print#1,a$;
```

```
130 print#1,tab(20);"xx",,"yy";
140 print#1:close1
150 rem"then use a$ as desired elsewhere
```

However, the B128 ROM routines handling this function have two or three bugs:

1. In line 110, the cursor position is not the same as it is for the PET, CBM & C64 machines, when prompting for input.
2. In line 120, the cursor and printing position of a$ is too far to the left, overwriting the last character of the input a$ which was already written to the screen.
3. In line 130, rather than harmlessly passing over an characters that may exist on the display, TAB(x), SPC(x), and the positioner comma (,) all destructively erase everything in their path.

For these reasons and others, the little routine shown above is not as useful as it might appear. I am sending CBUG a separate article describing in more detail the advantages and disadvantages of such an INPUT# routine, and showing a short but improved routine that can be used as a generalized replacement for the normal INPUT command.

8050 DRIVE CONVERSION TO DOUBLE-SIDED: There once was a persistant rumor that the Protecto 8050 disk drives used the DOS 2.7 of the dual-sided 8250, and that they could be upgraded to 2-megabyte 8250 status by "merely" changing the drives to dual-sided units, MPI 102's instead of MPI 101's. I've been unable to contact MPI or Shugart or other possible dual-sided drive replacement makers, and haven't been able to locate people or articles concerning the conversion. Was it just a rumor generated to help sell the 8050 drives?

TRANSBASIC FOR THE B128: The Canadian TRANSACTOR magazine has had an excellent series on TRANSBASIC, an endless series of pick-and-chose machine language enhancements or expanded BASIC for the C64. Is there any chance that someone could covert these for use on the B128? Many are supposed to be ROM-independent, others should convert mostly by address changes.

BUG - VERTICAL GRAPHIC CHARACTER MIXUP: Notice the neat or logical layout of the vertical and horizontal line graphic characters on the sides of the keys of your B128 keyboard. The leftmost vertical line character starts with a "control-5" from the top key row, and the lines progress to the right as the user key presses progress down the key rows to the "shift-n" and "shift-m" and counterclockwise back up to the top row "control-6". Likewise, the horizontal line characters progress from the top row "control-4", down to the bottom row "shift-c" and "shift-m" and back up to the top row "control-6". Or do they? Switch to Graphics mode, press down the shift lock key, then instruct your B128 to PRINT"DCVF". Note that the resultant characters are NOT in descending order as the key labeling state that they should. This could be corrected merely by re-labeling the keys. However, that leaves the key arrangement out of the logical or symettrical order on the keyboard, suggesting that the error may have originally crept in during the original assignment of ASC and CHR$ codes. This could be fixed with a new character-generator ROM, but I imagine that the machine would then be incompatable with other B128s or B128 programs.

DISK LIBRARY ORGANIZATION: At the moment, most of the CBUG disks tend to be clearly structured "Collections" by and from individual users. As people locate or submit individual programs however, there will be disks filled with programs having various or random applications. Most people would probably agree that users groups and individual users need to divide their libraries into two distinct yet duplicative catagories - disks of randomly mixed programs "as submitted" (the CBUG PRE-RELEASE series), and disks of mostly the same programs, after catagorization and debugging (the CBUG ARCHIVE series). The subject catagories used for the "archive" disks tend to have standardized somewhat, through the years of experiences of TPUG and other user groups. I generally agree with TPUG's "industry standard" catagorization, but would like to put in my vote for a slight expansion their grouping, mostly to reduce the volume and searching associated with some of the large, catch-all catagories such as "Utilities" and "Math-science", provide for a "bug-swatter and help source" disk, and keep ambitious single-user "collections" intact. Whether for a user group or for individuals, it is important to make an organized library plan and set a consistant method for assigning disk identifiers. For libraries that show promise of getting quite large, I vote for a catagorization plan something like the following:

I.D.	Subject Catagory	Catagory Explaination
ax	assembly languag	tools & utilities, FOR m/l-a/l programmer
bx	business & home	serious non-tech general applications
cx	communications	everything pertaining to modems,bbs,etc
dx	demos & graphics	non-interactive demos and displays
ex	education.xxxxx	student & teacher, subcatagory per subj
gx	games, text only	fun only non-educ programs,no utilities
gx	games, graph&mus	fun only non-educ progams, no utilities
hx	help & bug swats	bugs,solutions,parts,repairs,addresses,
lx	languages.xxxxx	computer languages, subcatagory per lang
mx	math, statistics	pure, general-purpose math,stat programs
nx	engineerg,scienc	applications for specific technical areas
px	program routines	subroutines for use in BASIC & ML progrms
rx	reviews,prgs&equ	reviews of programs & equipment,w/subcats
sx	sounds and music	subroutines for use in BASIC & ML progrms
ux	utilities.disk&t	tools to handle,analyze,fix disk or tape
ux	utilities,comput	tools to investigate,manipulate, computer
ux	utilities.progmg	tools to write,analyze,fix BASIC programs
ux	utilities.prntng	tools to format,manipulate printrs,plotrs
ux	utils. other i/o	tools for joystiks,paddle,voice,other i/o
xx	best of catagory	best of each catagory, updated once/year
zx	collections.name	ambitious personal collections/user name

Art Klinger, 4659 Balboa Drive, Wichita Falls, Tx.

PAID ADVERTISEMENT

```
*****                       *           *
*    *                      *
*    *     ****     ****    *          **     * ***      ****
*****     *    *   *    *   *           *     **   *    *    *
*    *    ******   ******   *           *     *    *    ******
*    *    *        *        *           *     *    *    *
*****      ****     ****    ******    *****   *    *     ****
```

If you've been looking for a terminal program that's easy to use, powerful, flexible, and thorough, then your search is going to end at 'BeeLine'. LemData Software has designed this terminal program to be easy to learn as well as flexible for the experienced user. With two Main Menus providing the overview of all options, it is easy to select your function and follow the prompts to complete your selected task.

BeeLine can use internal RAM or a direct Disk file for Buffering your 'ONLINE' data. Buffer storage is at least 75K and perhaps 200K if you have expanded your B-128 to 256K of RAM. With direct file buffering you can fill an entire disk, equal to 520K of data. The RAM Buffers can be viewed using various options, like a 'GOTO' function to begin viewing from any point in the buffer. You can block a portion of the Buffer for outpputting to your printer or disk. Disk files loaded into a Buffer may be in place of data already there, or appended to the end of the current Buffer data, appended to the Buffer from the end of the current block. Any type of disk file can be loaded into the Buffer including a Relative type file.

BeeLine has a phone directory of 20 phone numbers and their descriptions. These can be auto-dialled if your modem has that capability. When dialled, the phone number with its description is stored in the buffer for later reference. Any number of these directories can be saved to disk for loading later when needed. You may obtain a hard copy of your current phone directory by outputting it to your printer.

The B-128 function keys have also been implemented. This means they can be programmed to send your logon sequence or 'ONLINE' signature. It is even possible to program certain command sequences for BeeLine. As with the phone directory, you can save any number of function key files to be loaded and used as necessary. Likewise, the current function keys may be output to your printer.

BeeLine provides a character conversion table to redefine each character transmitted or received. The default table provides a clean ASCII conversion to CBM functions in order to prevent spurious codes from affecting your screen operation. Of course, you can define your own character set to respond according to your needs.

BeeLine can transmit a data file from your disk while you are 'ONLINE'. This will allow you to write your electronic mail offline and transmit it from disk when you need it. Two options for transmitting are provided. You can send just one line at a time from the disk file, pressing the space key when ready for another line of data. The second method is to send continuously using XON-XOFF control characters to prevent overrunning the host system.

BeeLine will allow a Baud Rate between 300 and 9600, 7 or 8 bits per character, and Even, Odd, Mark, Space, or No parity. In addition, you can specify an echo mode of Local, Remote, Local & Remote, or None.

BeeLine has the ability to save many of your specified parameters in a SETUP file on disk. The SETUP file on your BeeLine disk will be loaded when the program initializes. In addition, you can save a SETUP file on other disks, perhaps one for each of the information services you use. Simply insert the appropriate disk, load the new SETUP file, and your program parameters are set.

LemData Software will support BeeLine with additional features in the future. Don't be left behind.

Now Available For Only $35.00 - Order Yours Today From CBUG

BeeLine is Copyrighted 1985 by J & K Lemkelde - LemData Software, P.O. Box 175, Dover, PA 17315

CBUG COUPONS

YELL FOR HELP

This is one of the most important functions of CBUG. Most B128 owners are using their machines in business applications. To be down or locked up with a problem is intollerable. Picture the minister who accidentally scratched his Sunday sermon, or the businessman who trashed the directory of his Accounts Receivables disk (so the disk would not read no matter what he tried). These are both true examples and CBUG members have come to the aid of each other thru referals being provided by phone. We need more help in the field for these persons. Many callers are just in need of ongoing hints on program application or odds and ends input about the machine or quirks many of us have solved, others in dire straights.

PLEASE, PLEASE, if you have expertise in some area, register and indicate areas of expertise on this coupon. It is really a very rewarding experience.

STUMP THE AUTHORS, OR WE WANT YOUR INPUT

The CBUG publications are here to help you, all of you. A mite impractical of a goal I admit, but let us try. Recognizing that there is a huge range of interest and wide varience of levels of expertise, we need to know what each of you believes you need to see in CBUB- ESCAPES -- by subject, by level of advancement, etc. So as to try and simplify atleast some of this data gathering, we would like to resort to "mark the box" type answers. You are welcome to provide an essay type answer on a separate attached piece of paper, but too many of them will bog us down.

The boxes are for you to insert a number as to the degree of advancement you would wish to see in the article on that subject, with 0 being for beginners who have been unable to find meaningful help in the instrucion books..., to 9 for those who are already finding undocumented and advanced uses of the program.

After the box is a space for ONE brief word if you believe further explaination is necessary.

PLEASE mark only the four subjects of most interest to you.

PLEASE WRITE OR PRINT NEATLY. IT IS NECESSARY IF WE ARE TO CORRECTLY TRANSCRIBE YOUR INFORMATION

CBUG COUPONS

YELL FOR HELP

Only mark those in which you can offer help

Zip Code ______________

Phone # __________________________

Hours available: _______ to ________

Name ________________________________

Date ________________

	Pretty Good	Expert
Telecommunications	□	□
Instrumentation	□	□

	Pretty Good	Expert
Superscript II	□	□
Superbase	□	□
Calc Result	□	□
Word Result	□	□
CABS	□	□
Basic	□	□
Machine Language	□	□

NAME ____ ZIP _____________

A. SOFTWARE ARTICLES:

1. UTILITIES	□	__________
2. GAMES	□	__________
3. PRINTER INFO	□	__________
4. BUSINESS	□	__________
5. DISK APPLICATIONS & INFO	□	__________
6. PROGRAMMING - BASIC	□	__________
7. PROGRAMMING - MACHINE LANGUAGE	□	__________
8. EDUCATIONAL APPLICLATIONS	□	__________
9. OTHER SPECIALTY ________________	□	__________
10. GENERAL HELPS	□	__________

B. APPLICATIONS ARTICLES:

11 SUPERBASE	□	__________
12 SUPERSCRIPT	□	__________
13 CALC RESULT	□	__________
14 WORD RESULT	□	__________
15 CABS/INFO DESIGNS	□	__________
16 CMS/SOUTHERN SOLUTIONS	□	__________
17 TELECOMMUNICATIONS	□	__________

C. CO-PROCESSORS	□	__________
D. HI TECH PROGRAMMING DOCUMENTATION, ETC.	□	__________
E. DO IT YOURSELF SERVICE INFORMATION	□	__________
Z. AN EMULATOR TO MAKE YOUR IBM RUN LIKE A "B"	□	__________

CBUG ARCHIVE ORDER FORM
AND
NON-DISCLOSURE AGREEMENT

As a condition of receiving copies of CBUG Archive documents I agree and represent as follows without limitation or qualification:

1.) I am a member in good standing of the Chicago B128 Users Group;

2.) I intend to use the materials requested to enhance my and/or other CBUG members' usage of the B128 series of computers and accessories;

3.) I will not disclose, divulge, nor copy the contents of these documents to non-members of CBUG, except as they may be acting for me or on my behalf in the usage defined in #2 supra;

4.) I will cause a signed copy of this agreement to be forwarded to CBUG upon any further disclosure or release of these documents as permitted supra

5.) I understand that many of the documents covered hereunder are either copyrighted by or proprietary to Commodore Business Machines and/or their vendors; that the sole purpose of the provision of these informations by CBM to the CBUG membership is to enhance the usefulness of the CBM products purchased by CBUG members; and that there is no warrantee or assurance of accuracy, completeness or other representation by CBM, CBUG or any other. Any other disclosure is expressly prohibited, and I will not engage in such prohibited disclosures.

6.) I have read the limited no cost license (as printed on the reverse side hereof) extended to CBUG by CBM, and agree to be bound by those same said terms as if I were the named licensee.

Name____ Please print or type

Address____ Phone____________________

City__________________________ State_____________

Country__________________________

X____ Date______________________
Signature

Qty	Cat #	Archive Description		Price	Extension
_____	12370	001	8088 Schematics w/ CCPM listings (printed)(40 pgs prox) <Add $5.00 air mail if not USA or N. America>	$25.00	__________
_____	12365	002	Software Devel. Information (microfiche)(302 images/pages prox)	9.00	__________
_____	12351	003	CBMII Programmer's Reference material (3 microfiche)(798 images/pages)	27.00	__________
_____	12327	from pg _____ to pg _____ of Archive # _____			
		from pg _____ to pg _____ of Archive # _____			
		from pg _____ to pg _____ of Archive # _____ Custom produced photocopy of Archive documents.. Allow 45 days for delivery. Add $.05 per page for air mail if not USA or North America. $10.00/order minimum photocopy order		.08	__________

AGREEMENT MUST
BE COMPLETED
AND SIGNED
BEFORE WE CAN
SHIP ARCHIVE
MATERIALS.

DISK UPGRADE ORDER FORM

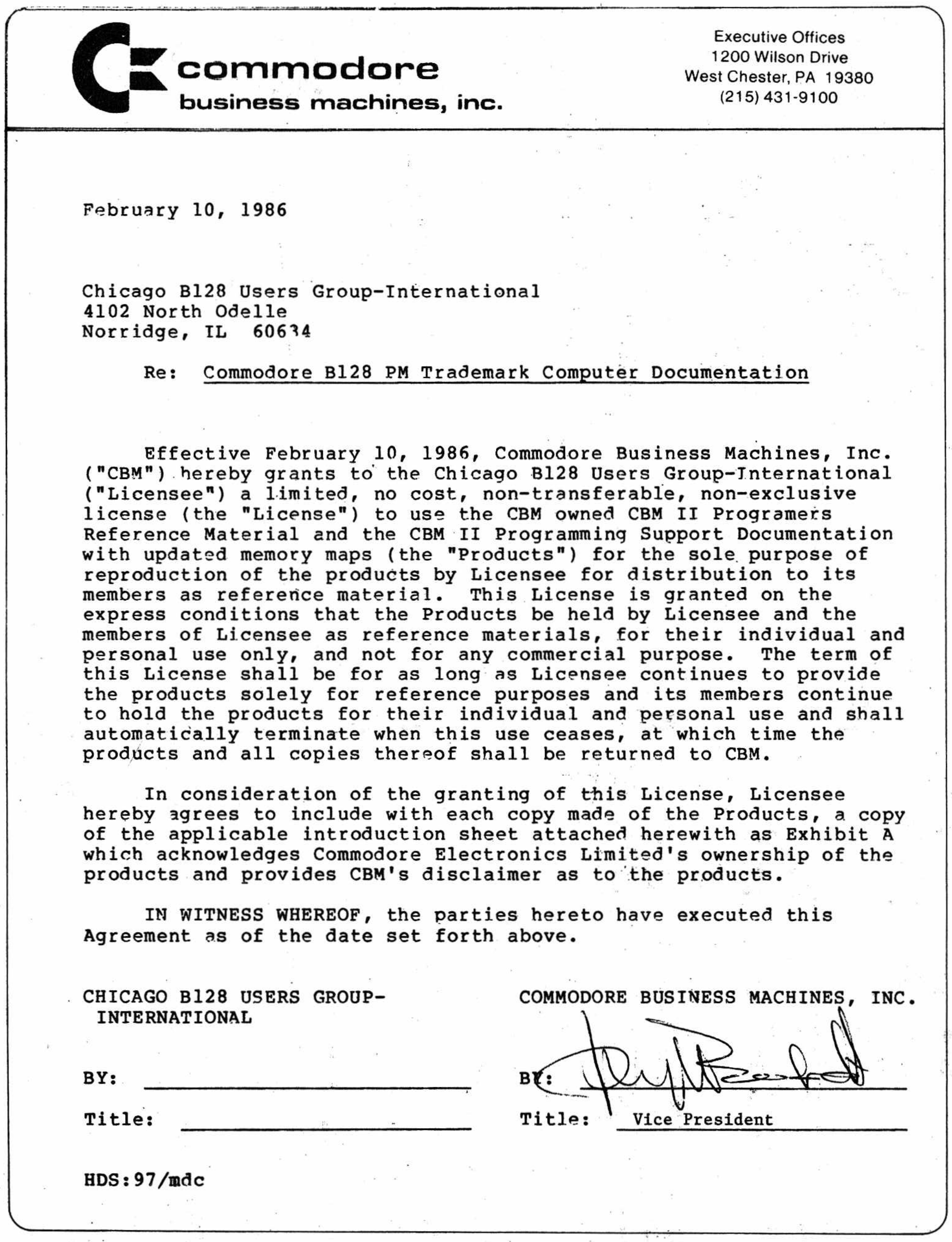
_____	12999	CBUG #17 LIZ's UTILITIES, v1.2 Ugrade purchase only	8.00	__________

AFFIX LABEL FROM YOUR EARLIER VERSION
CBUG STOCK #12877 HERE TO QUALIFY FOR
THE 50% UPGRADE DISCOUNT.

TOTAL ORDER THIS PAGE $__________

LEGAL NOTICE

BELOW ARE THE TERMS AND CONDITIONS IMPOSED BY COMMODORE AS TO THE RELEASE OF B128 DOCUMENTATION TO AND THRU CBUG. PERSONS ACQUIRING THIS INFORMATION DIRECTLY OR INDIRECTLY ARE HEREBY NOTICED OF SAME



commodore
business machines, inc.

Executive Offices
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

February 10, 1986

Chicago B128 Users Group-International
4102 North Odelle
Norridge, IL 60634

Re: Commodore B128 PM Trademark Computer Documentation

Effective February 10, 1986, Commodore Business Machines, Inc. ("CBM") hereby grants to the Chicago B128 Users Group-International ("Licensee") a limited, no cost, non-transferable, non-exclusive license (the "License") to use the CBM owned CBM II Programers Reference Material and the CBM II Programming Support Documentation with updated memory maps (the "Products") for the sole purpose of reproduction of the products by Licensee for distribution to its members as reference material. This License is granted on the express conditions that the Products be held by Licensee and the members of Licensee as reference materials, for their individual and personal use only, and not for any commercial purpose. The term of this License shall be for as long as Licensee continues to provide the products solely for reference purposes and its members continue to hold the products for their individual and personal use and shall automatically terminate when this use ceases, at which time the products and all copies thereof shall be returned to CBM.

In consideration of the granting of this License, Licensee hereby agrees to include with each copy made of the Products, a copy of the applicable introduction sheet attached herewith as Exhibit A which acknowledges Commodore Electronics Limited's ownership of the products and provides CBM's disclaimer as to the products.

IN WITNESS WHEREOF, the parties hereto have executed this Agreement as of the date set forth above.

CHICAGO B128 USERS GROUP- INTERNATIONAL	COMMODORE BUSINESS MACHINES, INC.
BY: ______________	BY: ______________
Title: ______________	Title: Vice President

HDS:97/mdc

CBM II PROGRAMMER'S REFERENCE MATERIALS

CBM II PROGRAMMING SUPPORT DOCUMENTATION

COPYRIGHT:

This document is copyrighted and all rights are reserved by Commodore Electronics Limited. No part of this document may be duplicated, copied, reproduced or distributed in any form or by electrical or mechanical means, including information storage and retrieval systems, without permision in writing from Commodore Electronics Limited. Duplicating, copying, selling or otherwise distributing this document is a violation of the law.

DISCLAIMER:

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PRELIMINARY IN NATURE AND IS SUBJECT TO CHANGE WITHOUT NOTICE. THE INFORMATION CONTAINED HEREIN IS SUPPLIED IN AN "AS IS" BASIS, AND THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY IS SOLELY WITH THE BUYER. COMMODORE BUSINESS MACHINES, INC. ("CBM") MAKES NO WARRANTIES EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE ACCURACY, COMPLETNESS, QUALITY, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OF THE INFORMATION HEREIN CONTAINED. IN NO EVENT SHALL CBM BE HELD LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY USE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN, EVEN IF CBM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME LAWS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

HDS:98/mdc