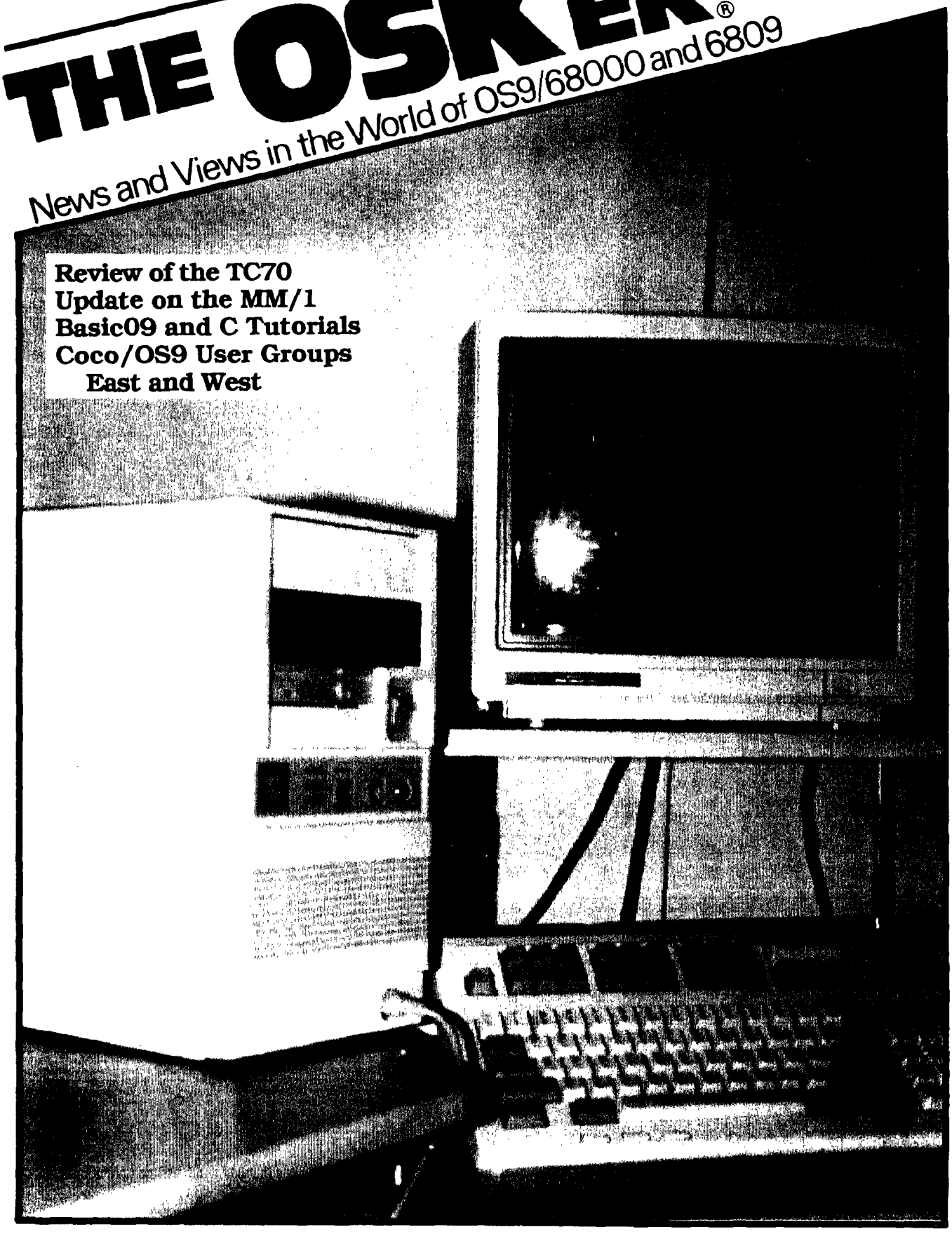
$3.00

THE OSK'ER®

News and Views in the World of OS9/68000 and 6809

Review of the TC70
Update on the MM/1
Basic09 and C Tutorials
Coco/OS9 User Groups
East and West



The Time Is Right.

The Time Is Right For Multimedia Δ

Interactive Media Systems, Inc. designed your MM/1 to provide you with a platform for now and for the future. Everyone agrees: multimedia is the future.

Multimedia standards are essential. That is why IMS has spent two years researching these standards, contributing its time on the ANSI HyTime multimedia committee, discussing multimedia clipboard options, and manufacturing the MM/1 with built-in graphics and sound. With a standard hardware and software architecture, programmers know exactly how to program to bring you the best in easy-to-use, colorful multimedia software.

The Time Is Right For Multitasking Δ

Your MM/1 comes with OS-9/68000™ Version 2.4, a pre-emptive multitasking operating system designed from scratch to be efficient, fast, and productive. And your MM/1 comes with a windowing system from Kevin Darling. OS-9 with windows ensures a wealth of applications and high performance.

Whoever You Are! Δ

Are you a hacker? You'll love the MM/1's Inter-IC interface, a 100,000 bits/second connector that supports voice synthesizers, digital filters, VCR programmers -- Signetics has dozens of options! You'll enjoy playing with the MM/1 video signals, SCSI interface, scanners, touch-screens and more!

Not a hacker? Get started with our User Guide. Enjoy built-in word-processing and graphics editor. Get on-line fast with Sterm. And there are plenty of applications for the MM/1, at great prices! Spreadsheet, databases, accounting. Just ask!

Save Money Today Δ

IMS, Inc. respects your pocketbook. You can get a simple-to-assemble MM/1 Extended in convenient kit form at a breathtaking price. Included is thousands of dollars worth of software -- no extra charge! **Call today and reserve a system!**

Specifications

CPU/Graphics: Signetics 68070 at 15MHz with 2 channels of Direct Memory Access, on board serial port and watchdog timers; Signetics Video System Controller 66470 with on board Run-Length Encoded graphics decoding, NTSC (TV) compatible sync rates for cost-effective desktop video publishing, and resolutions including: 320 x 210 (256 colors at once), 320 x 420 (256 colors at once) 640 x 210 (16 colors), 640 x 420 (16 colors), 720 x 420 (16 colors), and 720 x 560 (with multisynch monitors)

Palette controller: Brooktree triple 8-bit DAC for a choice of 16.7 million colors and maximal memory conservation

Memory: 1 Megabyte standard; field-upgradable to 3 Megabytes; write for info on 9 Megabyte upgrade

Input/Output: Up to 5 serial ports (3 standard); two bidirectional parallel ports; one serial port configurable for MIDI; one serial port powered for mouse; Signetics Inter-IC port for 100Kbps serial network or support for dozens of Signetics Inter-IC compatible chips; SCSI host adapter to support hard disk drives, tape drives, and digitzers (scanners coming in 1992); two channels of Analog-to-Digital Converters for sound or data sampling at variable sample rates up to 100KHz; two channels of DMA Digital-to-Analog Converters for smooth sound output without CPU intervention; Joystick port for Tandy 1000™ and Tandy Color Computer™ style joystick; Floppy controller built-in, with one 1.4 Megabtye Floppy Disk Drive included; RGB-A video at 15.75 KHz through standard DB-9 output (directly compatible with many popular monitors); standard PC-XT keyboard interface, either from on-board connector or 5-pin header for custom configurations.

Software: OS-9/68000 Version 2.4; C compiler; BASIC; Sequential Block File Manager (required for tape backup); PC File Manager (permits reading and writing of PC disks); Print spooling software; electronic mail (e-mail) for networked and multi-user systems; Sterm XModem and CompuServe-B protocol support; Paint™ package; Emacs 3.9 text editor; Proff text formatter; sound and graphics conversion utilities; Maze program; Tetrix game; windows for tilable, multi-screen windowing; other public-domain utilities; OddJob scripting language (*MM/1 EXCLUSIVE!*) supporting logic-control structures, interprocess communication, and other UNIX™ and awk-style functions; installation software; drivers for hard disk drives, floppy drives, parallel port, serial ports, mouse, realtime clock, and more.

Kit Includes: 1 Megabtye system with all the above (**$975**); Six-disk software set; User Guide; Kit Instructions; all cables needed (see options below); 1.4 Megabtye floppy disk drive; CompuServe™ SnapPak for free CompuServe time for new users

Options: Custom floppy and SCSI cables (call for quote); video adapter cable for Tandy CM-8™ ($35), two life-time-warrantied 1 Megabyte SIMMS for full 3 Megabyte operation ($150); IMS-approved slim-line PC case, with custom back-plate and pre-drilled for the MM/1 Extended kit, including 200-watt UL-listed and FCC-approved power-supply ($125); a variety of extension cables and adapters. Call for catalog

Interactive Media Systems Incorporated

Sales and Marketing • 1840 Biltmore Street NW Suite 10 Washington DC 20009 • 202-232-4246 • 3pm - 6pm

Call to place an order to save your place in line (small refundable deposit required); pay balance when you want; allow 30 days for shipping; write for information

the OSKer

issue #6

First look at the TC70..... 4
Review of the Tomcat70 7
Ramblin' Man 9
This is Glenside CoCo Club..... 12
The MM1's are coming! 14
Introduction to BASIC09 20
Review: StG-NET 23
Basic vs. C..... 27
M6809 Emulator..... 31

A first look at Frank Hogg's TC70 (Page 4)

'the OSKer' is published monthly or as possible by StG Computers inc., P.O. Box 24285, Speedway, IN 46224; phone (317) 668-8878. The president and editor is Scott Griepentrog, Secretary and sometimes columnist Chris Swinefurth, and treasurer Dave Henk. The position of V.P. is currently open.

Subscriptions to the OSKer are $12 within the U.S., $15 for Canada, and $20 elsewhere and overseas. Back issues are available for the cover price. Issue #1 is currently out of print.

Editing is done under OSK on a prototype MM1 using uMacs and also on a Tandy Model 102 Portable with Disk Video Interface (80 col display and floppy). Formatting and layout are handled by Word for Windows on a 486 machine (egads). Printing test copy is not possible due to the way the stupid software works, but final copy is done on whatever Apple LaserWriter (or other postscript printer) is within range. Zog's handling the printing, so I'm not sure yet how the magazine is duplicated. But it should look pretty good...

The entire contents of this publication are copyright by StG Computers inc., with the exception of entire programs or program segments printed herein, which are hereby placed in the public domain for free use by readers. Copies of articles may be duplicated or reprinted only by prior arrangement with the publisher.

To prevent a conflict of interest, StG Computers inc., as both publisher of the OSKer and having ownership of software, will not directly advertise in this magazine. Nor will the editor promote said software via his position.

The OSKer is to be an open and unbiased forum for all OS-9 (6809 and 68000) hardware and software. Any comments or complaints should be submitted in the form of a "LETTER TO EDITOR", which will be printed in the Editor's column. The editor reserves the right to print any letter sent to the OSKer (unless specifically asked not to), and also the right to edit or omit letters when necessary. However, any well-written complaint will always be printed, with any changes documented. The Editor wishes to convey as many points of view as possible to the readers, while striving to print only the truth.

Persons interested in submitting an article for publication should send it in ASCII text fomat on any OS9 or IBM disk format, or via electronic mail to CIS: 72427,335, DELPHI: TREVNICK, or StG Net: Sysop@Zog.. Articles selected for publication will entitle the author to 6 months of the OSKer for FREE. All submissions become the property of StG Computers inc. The deadline for material to be included in any issue of the OSKer is the last day of the month previous to the issue in question.

Cover artwork (and production work) by Alan Sheltra.

FIRST LOOK AT THE TC70

by Paul Pollock

This review, is one man's opinion, not to be taken as gospel. But for the purposes of academic reference, I'll review the reasons for what follows as conclusions. I've worked as a computer hardware professional for over ten(10) years. The company(ies) I've worked for have produced many different busses and architectures over the years, and I've had to service and repair all of their production systems.

These ranged from microcontrollers using MC6803's to monstrosities employing the IMP-16 minicomputer. In popular chips, I've serviced IBM XT-AT, and Motorola buss hardware using everything from the aformentioned MC6803, to MC68010. I've used everything from the obsolete Superbrain CPM computer up to the NExT Computer. I date back to the SOROC terminal, and the OHIO Scientific systems <grin>. In other words, I'm a hardware anachronism; meaning I've seen, tested or used almost everything in micro's at some point. And I've serviced a high percentage of these systems.

So it was, with some trepidation, that I accepted Jim Sutemeier's invitation to not only help with getting the new arrival running; but also to test and play with the system.

I have to admit at the outset, that I've been following Frank Hogg's progress on this system for almost a year now, so I was eager to see the results of Frank's efforts. And just between the reader and I, I personally like the TOMCAT design better than competitor products (DELMAR, MM/1, etc); especially as it regards the K-BUS interfacing specifications.

Jim had just called me Saturday, August 31, to tell me he was on pins and needles about the arrival of the 'beastie'. Then, the following Monday, at 10:30am, I got another call; "...hey Paul, wanna play with a new computer?..."! This of course, haralded the arrival of the 34 pound 6 ounce new born package; delivered by UPS stork service. Shortly after I arrived at his home, Alan Sheltra also arrived according to invitation; and we set to work.

The minitower case had already been removed from the packing, and was sitting (gleaming off-white, in the noonday sunshine, streaming in from the plateglass doorway) on the table, alone and somehow lonely. Next to it, sat the optional keyboard with the rollerball cursor control unit built-in. And strewn around the table was the almost ten(10) pounds of paperwork and manuals from Frank Hogg and Microware, and assorted other suppliers for such things as the minitower case and keyboard. And sitting on the kitchen counter, in a place of high honor, was the new hard drive (a Quantum 105slp), awaiting installation into the cabinet.

Firstly, we unshipped the minitower frame from the outside case, revealing the interior. Mounted high and towards the rear, was the 200 watt power-supply. And mounted to the main support chassis member (on 1 inch standoffs) was the TC70 computer card itself. Connected to it were several cables leading in all directions. One to the 3.5 inch 1.44 meg floppy drive, already mounted to the disk drive bay. Others to the front-panel mounted keyboard socket, the rear-panel mounted serial and parallel ports, and monitor output. All the last were 9-pin DIN affairs except for the DB-25 receptacle for the printer connection. All are wired in IBM AT type configuration, so off the shelf standard IBM cables may be used for standard function peripherals.

Then mounting the hard drive to the drive-frame, and installing the 50-pin ribbon cable to the drive and TC70 dip-header; we prepared to apply power. Alas! The hard drive refused to light the drive led, and refused to initiate properly. We removed the drive, and scrutinizing it, found the problem quickly. For SCSI hard disk drives, the drive select is normally done by a set of three(3) pairs of staking pegs which are pegged in binary addition for drives 0 thru 7. The Quantum came pegged from the factory as drive 6 for MAC compatibility. Yanking all select pegs set it for drive 0, and we tried again. This time, all went well, and we got a satisfying boot from the floppy, and the hard drive began to respond to commands (hooray!).

Then we put the whole cabinet together, carried the whole thing over to Jim's computer workstation, and prepared to set it up for perminent installation. Again we ran into a problem! This time, upon power-up, the keyboard refused to respond. Opening the cabinet, we discovered an intermittant in the IDC connector from the keyboard socket to the TC70 card. Some fast 'tweeking' and we tried again. Success! Then we checked video and found it dimmer than it should have been, andcould not get RED video at all. Again we checked cables, and found another intermittant in the DB-9 connector to CM-8 cable adapter at the IDC connection. Some more tweeking restored the RED, but the picture was still dim. A call to Frank Hogg informed us that we had to flip a dip-switch on the TC70 card to alter the video. Once done, video sharpened up alot, and contrast improved.

Attempting to use the serial port, detected another problem. This one proved to be a tough nut. Niether harness, hardware, or software could be found to be defective, yet no amount of prodding yielded more than a squeek from the modem. Back to the phone and help from Frank Hogg. At first, even he was stumped until he asked us what we had in the bootfile. Apparantly there are two descriptors that point to the serial port. One for communications service, and the other for printer service. And when both are iniz'ed (such as at bootload) these confuse the system and cause the port to be inoperative. We yanked the offending descriptor, and again success.

After all this, we began to have some real fun! The computer runs extremely fast, in comparison to Coco3 OS9 Level-2 operations. Complete bootstart required less than 2 seconds for load and startup shellscript execution. Prompt comes up with almost no waiting. Neato! My Coco3 with Diskmaster and 25 line startup file (including starting up TShell, loading fonts, etc); takes about 35 seconds. Geez Louise!

I began to understand all the excitement when we started doing file copies with 150Kbyte buffers, and I really had fun when I booted BASIC with 'BASIC #400' (that's 400Kbytes to us non-OSK folks), and it took! Not only did it accept the large memory call, but BASIC (which is twice as large a program as the OS9 Level-2 variety) answered up and signed on in a small fraction of a second!

DSAVE and DCHECK works effortlessly, as well as sample tests of programs like AR to crack archives. At least 10 times as fast as 6809 system software. Some functions operate so quickly they defy benchmarks, in Color Computer OS9 terms.

I suspect some of the increase is from the much more efficient hard disk; but most of the increase is the bigger processor, and more robust operating system software. In fact, it's easy to notice that a 40 track disk drive (which I also installed, via the addition of an additional card-edge connector to the existing ribbon cable) runs faster on the TC70 than a SCSI Hard Drive does on a Coco3 <great big grin>!

All standard software and utilities worked without a hitch. And it was a pleasent surprise to find out all standard utilities work the same as we are used to, and many have additional talents, or features which equate to more user control over the system (DIR auto-sorts the output, as an example). All utilities have helpfiles built-in. Another pleasent surprise is that all command extensions and parameters are fed via a '-' prefix in a consistant and predictable way (whereas in Level-2, parameters are fed by a number of non-cooperative methods, meaning the Level-1,2 user needs to become familiar with each programs' specific features; especially common in 3rd party tools and programs).

The keyboard is very nice, has an excellent feel, and an easy touch. Looks like it will prove durable, and accurate for the forseeable future.

The 3.5 inch floppy, seems to be a 1.4 meg Hi-density disk drive, although it is used in lo-density mode for system software diskettes. This is one of the few TC70 specific hardware areas discussed by included documentation. Although very sparesly. My guessing about the Hi-density modes are exactly that, based on clues in the single sheet of discussion included.

The computer card proper, is a beautiful piece of engineering. The folks at Hazelwood have done a great job of implementing FHL's plans and specifications. Everything on the main board is laid out in an orderly and predictable fashion, and the materials selected are of the highest quality. No cludges here!

Frank Hogg has apparantly refused to make use of the ports in the 68070 as there are chips on the board for parallel port (MC68B21) and the two serial port chips (MC68861), as well as on board floppy controller, SCSI interface; and Video System Controller. Either that or the onchip ports are relegated to non-essential porting operations.

The 1.5 megs of dynamic ram is mounted in the bottom of the PC-card, to facilitate cooling of these chips, as everything else seems to run cooler. This is smart design as the ram is often the hottest components in a computer.

And the battery for the real-time clock is a removable watch battery in a clip assembly, so it should prove user replacable.

Hardware dip-switches are provided to accomodate user selectible options for boot drive, as well as monitor types (CM-8, RGBi, VGA, etc); and terminal port.

In short, a modest OSK system (huge by Color Computer OS9 Level-2 standards) can be assembled with only this single card; yet because it is buss compatible, can be expanded to truly heroic proportions. It is totally compatible with KBUSS memory and port cards, so the limits are very hard to reach on this system.

Mini-Tower case seems to be made fairly nice, and the wiring is logical, clean, and easy to manage. Shouldn't be too tough to figure out, even for the screw-driver neophyte <grin>. Power-supply is an uncommonly small package for the 200 watt rating. Even so, it seems more than adequate for all requirements, and as a bonus the fan is reasonably quiet! Indeed, the whole package is so quiet that the computer makes no noise at all during operation. Not even the hard drive can be heard!

After the install of the HD, I deduce that the 'Turbo' mode lite wire harness could be used as an additional hard drive light if it becomes necessary.

Documentation!

This is a direct comment to Frank Hogg, and it should be dealt with before too many customers receive this system. TC70 documentation, dealing with TC70 hardware/software specific to this computer are nearly non-existant! Some of the custom utilities are discussed, but important features of system modules and terminal specifics are totally absent. There's no discussion at all, to deduce the usage of modules like /vt70 versus /term, etc. The fact that this computer contains one of the most powerful single-chip video effects processors available, isn't mentioned anywhere!

And since it is clear that OSK has very little application software available, it seems reasonable to assume that users will find they are writing their own programs initially. How does one take advantage of the system's unique features, without a complete manual discussion of screen dynamics, command primitives, graphics controls, terminal features, attributes; etc? There is no documentation of any kind. The user is initially left entirely at his own devices, trying to make use of this system for anything other than running utilities.

More, there's little or no description of specific termcap and termset features, which might be useful for users which need them. The files themselves are non-explanative. They need complete explanation as it relates to TC70. Microware describes them in general, but that's not useful as it regards a TC70 owner who's never even seen OSK.

In short, the system is a very pretty, fast running paperwieght; for anything like user applications. One can write utilities (since they don't normally need screen usage), but anything that makes real use of real computer features is absent (even if the computer and operating system supports tools a user might want). Microware did a terrific job with the Coco3 Level-2 video environment; and Tandy did an excellent job writing DOCs for it. The least that can be expected is that when a new system is built, specific special features are annotated correctly. In this system, they are not merely done wrong; they are (more or less) absent.

Closing Notes

None of the previous discussion is to be taken as derogatory comments. And it should be mentioned that this computer was sent out in a rush, since Jim Sutemeier confesses that he needled Frank Hogg almost daily to rush shipment of this unit. It is clear that in all the haste to get units into the field, the documentation has been left in the lurch. This is to be expected with a new product, and I feel sure that Frank Hogg will do his best to rectify this situation.

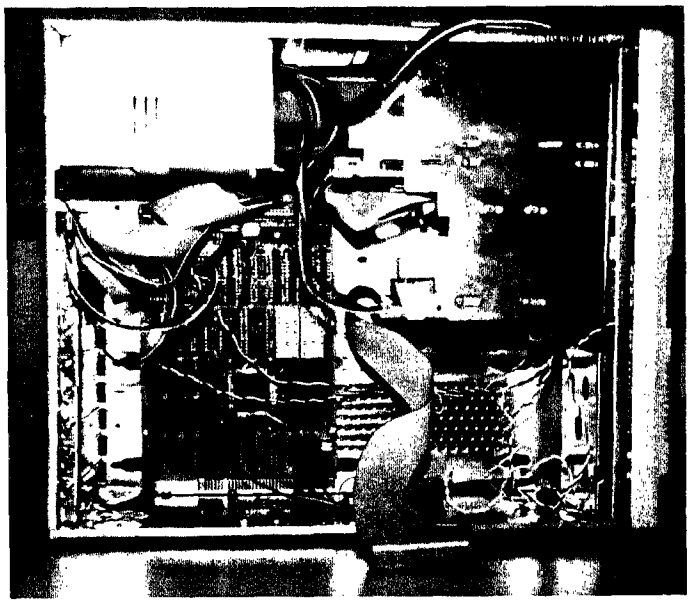
This is not to say that the TC70 is sent with no paperwork. It does come with a complete set of Microware Professional OS9/68000 manuals (although I found it strange that the system software came on 3.5 diskettes while the labels included were intended for 5 1/4 diskettes), and some paper bits and pieces required to explain some extroardinarily important software. In addition, Frank himself was easy to get ahold of to help in leaping over the small hurdles we encountered. This proved more than adequate for us to get the TC70 off the ground.

The previous lengthy discussion is meant as areas to be dealt with in future production. The TC70 is (on balance) an enourmous achievement, in terms of power versus price. It is fast, efficient, and nearly immune to system crashes. Hardware is (barring previous comments) solidly constructed, power supply seems stable and well built. Disk drive seems to be of good construction and operated according to available documentation. Languages included ('C', BASIC) provide the user with enough tools to use the system as a development platform. Documentation from Microware seems to be geared to their usual high (but cryptic) standards <grin>.

In my opinion, this system, with the advent of corrections to be made for previously mentioned problems and oversights; and perhaps the development of some nifty additions to the operating system, like extensions to make larger use of the extremely powerful graphic Video System Controller, could easily give MAC and ATARI a run for its money. This is not (as the TOMCAT label implies) a meek little housecat. It is a snarling, howlingly powerful little monster, that produces more power per dollar than almost anything else on the market.

Congratulations are in order to Frank Hogg, and a strong thank you to Jim Sutemeier for involving me in a 'first look' at this system!

TOMCAT TC70 'E' system, with AT-type keyboard and rollerball, and CM-8 monitor adapter cable; $1569 (approx); Frank Hogg Laboratories. Quantum 105slp Hard disk; price not available at presstime.



TC70'S innards revealed Mother Board at lower right, just below power supply.

Tomcat tm

THE TOMCAT9

Frequently call the "Coco 4". Run ALL of your current OS9 and RSDOS programs. Enhance your TC9 with the Tiger - a 68K interface that will speed up your TC9 by 2-3 times.

THE TOMCAT70

Capture the power of the 68070 CPU, linked with the power of OSK Operating System. Excellent graphics resolution. Fully expandable K-Bus with over 20 cards available now.

SIRIUS
SOFTWARE and HARDWARE

P.O. Box 153
Northridge, CA. 91328-0153
(818) 891-3369 (Voice)
(818) 894-0012 (BBS)

Review of the Tomcat70

by Jim Sutemeier

Let me start out by saying that I do understand some/many of the technical aspects of a computer, but not nearly enough to write any type of a real "technical" review of this new machine that I just received - the Tomcat70. I'll leave a tech report to those better suited for the job, and will only tell you what I think of this machine, based on my primarily 'user' status.

Let me give you a short background on myself - I bought a Color Computer back when they were just called 'Color Computer', about 10 years ago. I moved on to the OS-9 Operating System early on, thanks to a good salesman at my local Radio Shack, and have been 100% OS-9 for about 8+ years now. (Only RSDOS command I can recall is DOS!)

When Tandy dumped the Color Computer, and Microware stopped supporting OS-9/6809, I decided it was time to move on to a bigger and faster machine.

I looked carefully at all the advertisements for the 3 new machines, the System IV, the MM/1 and the Tomcat70. I also read all the messages on Delphi and Compuserve about these machines. (Sometimes you can get more out of messages than you can get out of ads!)

Well, I decided on the Tomcat70, as my choice. It's a little more expensive than it's competition, but, with the K-Bus, I can easily expand my TC70 into anything I want. I can add serial cards, up to 10 additional megs of RAM, even add a Tomcat9 card to act as a co-processor to the TC70.

Well, about 3-1/2 weeks after I placed my order, my Tomcat70 arrived in one, pretty-good-sized box. I, like a child with a $5.00 bill in his hand let loose in a candy store, dug into the box greedily.

Everything was there that I ordered. My Tomcat70 came equipped with 1536 Kbytes of RAM (standard equipment), and one 3.5" drive, which can be used in reg. density or hi density. I added my 40 track DS 5" drive, and a Quantum 105 meg hard drive. I ordered, in addition, the AT style keyboard with the trackball included so I wouldn't have to find a place for a mouse; also ordered the standard video to CM8 cable, the printer cable and the modem adaptor cable.

The hookup of all this equipment was easy and effortless. The biggest problem I had was trying to quell my excitement!

I got it all hooked up, turned on the CM8, and pressed the 'on' button. Well, it took about 15 seconds, and the system was booted! (Geesz, that's fast!!)

Well, it was time to take a trip around the machine, THEN I'll sit down and start reading. (Remeber, if it don't work, then read the docs.)

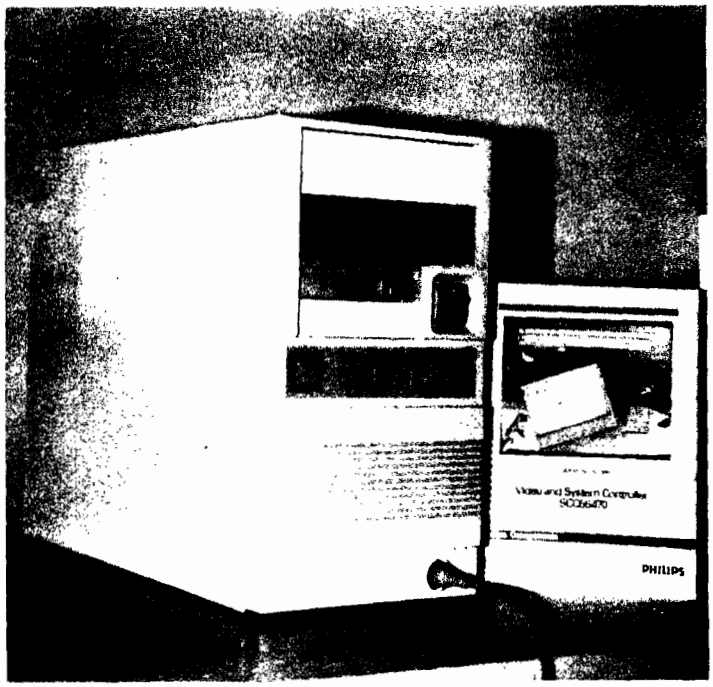
Well, there are 100 commands in the CMDS directory, plus a lot of directories full of stuff in here - excellent!

As I write in C, I first dove into the C/SOURCE directory, wrote myself a short routine that I had on my CoCo, and compiled it. The program took about 30 seconds to compile, a chore which took about 3 minutes on the CoCo!!

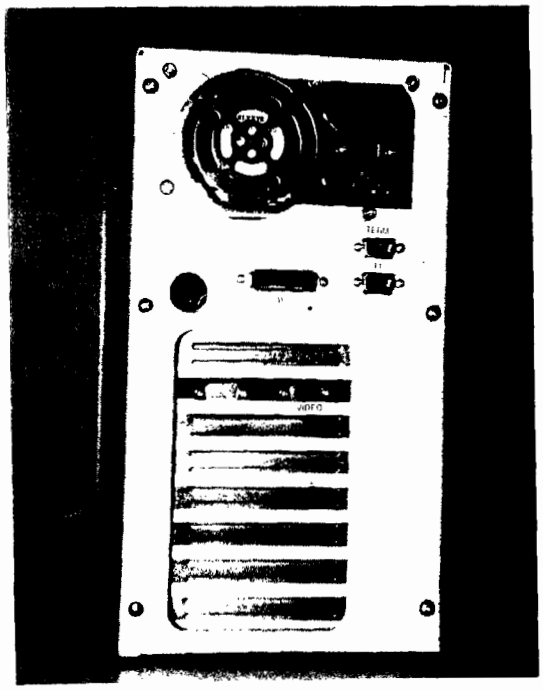
So far so good - let's try the printer out. Ahhh, very good.

Let's try out the modem -- whoops -- HEY FRANK -- got a problem with this modem - /t1 isn't recognizing my modem!! Well, a call to Frank Hogg produced immediate results - seems /t1 and /p1 (serial port) both use the same chip, and both were iniz'd - causing a conflict. Simple enough to fix - remove /p1 from the bootfile.

Ahh, now everything is working fine.



Comes with 1 3.5 1.44 Meg FD



Note: TERM, T1 and Parallel Ports

10 Days Later

Now have been using this new Tomcat70 system for about 10 days now, and am starting to feel very comfortable with this machine.

It is very fast and efficient. I have noticed some caveats in the C Compiler, but they are easy to program around.

I have tried out virtually every command in the system, and they work the way that Microware says they do.

The documentation that Frank Hogg and Microware supplied me is excellent and easy to understand.

Want List

Not included in this package is a windowing environment. HOWEVER, Frank has advised me that he'll make a decision on which set of windows he will carry (apparently he has a choice), and let me know very soon. (I'll probably have windows before you read this article!)

There is not a tremendous amount of available programs 'out there' for the OS-9/68K environment. While I may purchase (at very high prices) some programs, it seems there should be more PD type stuff available. I am hoping that with the influx of a lot of ex-CoCo programmers, that this situation will change soon, and there will be more PD stuff, and relatively less expensive software available.

Final Thoughts

I am really happy with this Tomcat70 System. It is packaged quite nicely in a Mini-Tower case, looks and runs real sharp.

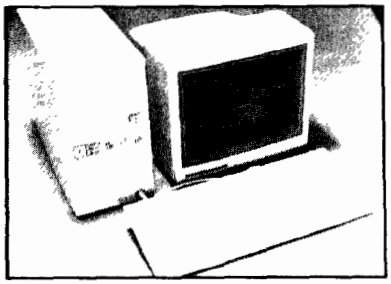
If anything ever goes wrong with this system, I KNOW Frank will be there to assist me. He's been in business for about 15 years now, and he stands behind his products 100%. I am not concerned that he might 'belly-up', and then I'll be stuck with something with no support. (One of my main reasons for purchasing from Frank Hogg!)

This system will keep me happily programming and enjoying for many years to come. At this point, the only expansion I can forsee that I might want is maybe a couple of more megs of RAM (2 to 4 megs), and the Tomcat9 card to act as a co-processor (that'll be fun to play with - the opportunities seem endless with that setup).

I would recommend this computer to anyone who has a desire to upgrade from an 8/16 bit environment to a much faster and more powerful 16/32 bit environment. (15 mHz is incredible!)

To someone who already uses a fast machine (clocking at say 10mHz or better), I'd suggest this machine, because then you'll have the OS-9 environment to work in. It's a UNIX clone, and is a very powerful real-time Operating System.

The Tomcat70 will allow you to expand your system, when (and if) you desire to do so.



Tomcat tm

The TOMCAT70, the 68K computer of choice for Tomcat/Color Computer/68K users. The TC70 is the latest in the line of K-Bus compatible products, providing the greatest flexibility and expansion for the OS9/OSK Community.

Signetics 68070 CPU @ 15MHz
Graphics Resolution 320x200 to 720x540
RGB-Analog for CM-8, RGB-TTL for PC's
DMA SCSI Host Adaptor for Hard Drives

1.5 MB RAM(1,536K)
16 to 256 colors on screen
DMA Floppy Disk Controller
TC9 Compatible

SIRIUS SOFTWARE and HARDWARE

For Further Information or Brochure, Call or Write:

Official FHL Distributor
P.O. Box 153
Northridge, CA. 91328-0153
(818) 891-3369 (Voice)
(818) 894-0012 (BBS)

*The Color Computer is a registered trademark of the Tandy Corp.

OS9 is a registered trademark of Microware Corp.

by Scott Griepentrog

You've probably noticed some changes in this issue. A little nicer looking, a little thicker, and actually readable. Well, as time goes on, I stop experimenting with different methods of producing this magazine and settle on what works. Last issue saw the use of a new DTP program to do the layout, namely Microsoft's Word for Windows. Well, you probably had difficulty reading the last issue too. That's because Word is not true WYSIWYG (what you see is what you get). And, although it supports many different printers, it does not support them the same way. What I mean is, I layed out the issue and tested it on a 24-pin printer. But when I told it to switch to a laser printer, it completely mangled my layout. No longer did the magazine fit nicely into exactly 24 pages, with columns just fitting in the space available (something that took me quite some time to do). It now took up about seven additional pages, with lots of gaps. The Word for Windows program figures the display and layout based upon the fonts available within the particular printer you have selected. Which means that if you intend to do a final copy on laser, there's no sense in even trying it out on a dot matrix first - it won't look the same. Well, anyways, I regret the readability problems. Call it on the job training. It won't happen again. I've started this time with Word set to output to a Postscript laser printer. Because Postscript is (more or less) the same on printers supporting it, I can get away with dumping the output file to any one of several I can get close enough to smell that toner...

Production and mailing of this issue (i.e. final inserts, printing, etc.) is being handled by our friend Alan Sheltra, who is a graphics artist by trade. As I am running very tight on available time (not to mention funds), I figured it made sense to let him do what he does best. I will be seeing the results about the same time the rest of you do (he's in California, I'm in Indiana). Actually, that reminds me of something. Although I haven't yet moved the OSKer post office box, I do have a new number to contact me: (317) 668-8878. The old number, (317) 241-6401, still works for the moment. It is being forwarded to the new one, and I would appreciate ya'll spreading the word about it. The reason for the change is because I've moved to Marion. Not exactly the best place to be, but because of some business contacts there it is necessary for the time being. I wonder if I can get a P.O. Box number like 6809 or something?

Also in this issue you will find a pair of articles about the new TC70 from Frank Hogg. I'll be able to get my hands on a TC70/TC9 pair in the near future, but it helps to get info out as soon as one can. Somewhere in here is an article about programming in Basic09 verses C, and a blurb about the Glenside Club, the most successful CoCo Club around. And, I have received the new software for the MM1, which I cover in detail, along with an update on the goings on in that camp. Oh, and I haven't forgotten about the M6809 review either...

If you're reading this, you're either at or have missed the Atlanta CoCo Fest (Oct 5-6). I've just now apologized to Dave Myers (CoCoPro, they put it on) for not getting the word out in time. The deal that was supposed to keep me in the pink for a while fell through when the guy decided not to pay me. For you fledgling programmers out there, a word of caution: ALWAYS GET IT IN WRITING! Never trust anybody, no matter how kosher they seem. Seems I'm destined never to follow my own advice. So, the last two months of my life (solid) have been dedicated to producing a new program (for data processing using MS-DOS) so I can eat the rest of this year. It was supposed to take me a month, but wouldn't ya know it, it took two. Ain't that always the case? Anyways, I just yesterday finished the first version, and am driving down to Tennessee to test it tomorrow. My calendar refuses to let me add a few extra days between now and the fest so I can get this layed out, printed, and mailed. That's life. And because I'm scraping the bottom of my wallet to cover the costs of printing this issue, I'm being forced to make some changes. But, I'm going to give you fair warning before I do. The subscription cost will be going up after the end of this year. That means you have three months to get your subscriptions in at the current $12 ($15 in Canada) rate. I really should double the costs, but I'm not sure if people would pay that (though it would mean getting this out more frequently). I would appreciate feedback on this question. Would you pay $18 (50% increase) or as much as $24 for a year of this magazine? It's still costing me more to produce than I'm getting back on it (will for some time), but I believe it's a good investment (when I have the money & time), and every little bit will help. You can send messages to me via USmail, CIS: 72427,335, or Delphi: TREVNICK (don't ask). While you're at it, let me know what you think of going on a 6 month/year schedule instead of a monthly one (I'm almost making that now ;->).

Well, it's time for a few words from our sponsors. I've gotten several positive comments about my review of the MM1 in the last issue, but it's the negative ones that are always more interesting:

> Dear Scott:
>
> First of all, let me say "Kudos on your exellent magazine entry to the world of OSK computing!" Now, to the bad news. I must say that you have exhibitted a lack of professionalism with your review of the MM/1 in Issue #5 of *the OSKer.*
>
> You start out by referring to Paul Ward as "the irascible Paul Ward." Now I'll be the first to admit that I do not have a copious vocabulary so I had to look up "irascible." According to my Websters it means "easily angered; hot-tempered." From then on in your article you put forth *nothing* to back up this specific comment of Paul Ward's character. In fact, judging from the tone the article, it seems to me that **you** are the one who is irascible.

Then, in this article which is supposed to be a review of the MM/1, you devote almost three columns to your personal opinion about IMS and exactly how you perceive they have treated you in various instances and circumstances. While I will certainly agree that IMS has made some mistakes in the past and that you may certainly have been the unfortunate recipient of some of those mistakes, a supposedly unbiased review of the MM/1 is no place for your personal opinions of IMS and their business practices. In the future I would appreciate such comments to be located in editorial comments.

I also take exception to some of your personal comments on the MM/1 which were obviously fueled by your personal feelings in the matter, specifically calling the MM/1 "the Mickey Mouse One." Totally uncalled for. When you finally get to the meat of the review, anyone with half brain would be able to determine that the MM/1 is, for the most part, a well designed computer worth the money it costs. Prefacing such a review with your negative comments based on personal feelings was inappropriate.

I would have no problem if your personal comments appeared in an editorial column but they didn't. They were included in a product review which should have been as objective as possible. In case you haven't guessed by now, I am a supporter of IMS, have my MM/1 on my desk as I speak, and am a member of the IMS Developer's Association.

Now, on to another issue. I would also like to address the letter from Mr. Hutchins. You title it "Why the 'CoCo 4' Will Fail," and indeed, he ends his first paragraph with "... I believe the 'CoCo 4s' will fail." He then goes on to spend two paragraphs on the shortcomings of available OS-9 based software with examples specifically in the OS-9/6809 arena. What has that got to do with the 'CoCo 4'? Especially since all 'CoCo 4's are 680x0 based systems.

Here he is obviously indicating an uninformed opinion. When I first got into OS-9 myself, I set about to write as much software as possible. As you well know, when you are working alone, it takes time to crank out quality code. The CoCo3 had only been on the market for a little over a year when I got my first one. I had Multi-Vue, the C Compiler and the Development Pack within weeks after that. I released my first OS-9 Level 2 program just a few months after that, Pyramid Solitaire. You may have seen it.

Since then I have written five other game programs, four other card game based programs plus a rather complex Word Processing Oriented Shell which I call WPShel and have gone commercial since Shareware returns on the early programs were meager at best. Even with advertising in *The Rainbow* I still did not make a profit. So, people who compliain about a lack of software bug me when I have been doing as much work as I have been to provide software and the market has not responded.

His negative comments about Word Processing software for OS-9 are also laughable, which I hope are due to his lack of knowledge. This letter was formatted on my CoCo3 using VPrint from **Bob van der Poel Software**, as was the enclosed catalog of my software. Yes, they look great and my laser printer helps a lot but most any 24pin printer can produce similar results. But the point is that *it was the software which made it all possible!*

Let me make another observation. I don't know if you took a good look at the actual copy you sent out, but my copy *the OSK'er* was so dark that it was very difficult to read.

Let me finish up in what you may consider a surprise way. I still think that you have a good thing going and would like to take part in it. I will most certainly be an advertiser in your next issue. I am also interested in contributing an article or more to your magazine. I have some ideas for material, but if you have any suggestions, I'm open. My forte is C programming.

Sincerely,

Zack C. Sessions

Proprietor, ColorSystems

Let me shake the kinks out of my fingers from typing in your letter for a sec.. <crack> Yah, that's better. Okay.

Lack of professionalism. Hmm. Yes, I would have to agree that one could easily take issue with my having mixed opinions with a review of the machine. I had originally intended to keep my 'comments' in the separate article about Paul and IMS, but somehow they got sucked into the actual review as well. It just might have had something to do with the lack of anything else to talk about. As I'm not trained in the art of journalism, I will gladly conceede to being called unprofessional. Hey, I profess to be a hacker, and that's about it. I'm sitting in the editorial chair because nobody beat me too it. If somebody wants to try it out for an issue that can be arranged. Do let me know. But I digress. Or maybe I just ramble. Never can tell.

I'll also have to agree that I could have done better on the word irascible. Yes, I know it means easily angered, and anyone who knows Paul knows that the last thing he is. Mr. Paul 'Cool Head' Ward, we'll call him. When I wrote that line in my head I said the word sarcastically, but of course, that seldom comes across well in print. I gotta remember not to do that anymore...

I had to go back and re-read that part about Mickey Mouse. I didn't think I had called it that. The way I worded it, however, it could certainly sound that way though. What I meant to say was, "But I did come by [a message from somebody who had] an appropriate nickname for Paul's Computer: the Mickey Mouse One." Not that this version is all that better, as in it I'm still sort of agreeing with the person by saying 'appropriate'. The same paragraph starts with "So, as IMS is obviously reluctant to let me examine their software..." I'll explain my reasons for this in a minute.

I have an MM/1 on my desk too (am typing this into it), and am a member of the IMS Developers Association. But do I support IMS? Well, I have a program available on both CIS and Delphi written for the unit. And I have no less than four or five projects in the works (some more than half written). As long as I can beat some response out of IMS (in whatever way it takes), I'll continue to write stuff for the machine.

I didn't title Jim Hutchins' article, he did. Minor point, I know. And I certainly don't agree with what he said (and said so). I was not going to print the article until someone suggested that getting rebuttals would be a neat idea. It gave Paul and Ed a chance to speak their mind. I think the entire column, as a whole, worked out okay. I could be wrong. However, I don't see how you consider all CoCo 4's to be 680x0 based systems. The TC9 is an excellent example of a machine that can be called a CoCo4 (some people in fact reserve the term exclusively for it).

Believe me, I feel exactly the same way as you do about the difficulties of so much as breaking even (even) in the OS9 market. If I were to total up all the money I've spent trying to sell or produce software, and subtract all the money I've made from same (cough, wheeze), I'd probably come up with about four, maybe five thousand dollars (okay, so that's over a couple years). And that's not including what I've lost so far on this rag. But, I have faith that one day (better be pretty darn soon now) it will all be worth it. Because I haven't sold out (ahem, well, not completely yet), I'll be ready to go when the rest of the world all of a sudden goes, "Oh my god, look at what they can do on that little machine!".

Yes, I know about the dark print. It was a wee little bit dark when it came off the printer (had just put in a new ribbon), but was still very readable. Unfortunately, the fact that the printing process darkens the text a wee little bit had not slipped my mind when I took it in to be printed. I didn't particlarly have much of a choice, short of doing it over. But hey, look at it this way, you didn't really want to read about me griping about this and that and the other thing did ya'?

I guess one has to keep one's tongue firmly implanted behind one's cheek at all times when reading this column. Goodness knows I have to in order to keep from choking on the stuff as I write it. Although goodness didn't have all that much to do with it. Did ya'll here the one where the guy puts in "Yes please" in the box marked sex on a job application?

Offering to write an article? Hey, watch it. I'll take you up on that one. How about continuing where I left off with the Playing Chess in C series? I sorta dropped that due to lack of interest. I'm not sure whether it was mine or the readers. But I'd be happy to print whatever you can send my way...

Well, out of the frying pan, into the fire:

> Dear Editor:
>
> Continue your hatchet job on your own dime. I hereby cancel my subscription to The OSKer, and would appreciate the return of whatever money corresponds to unreceived issues remaining.
>
> Yours truly,
>
> James Jones

Yikes. I will have to admit this one got to me. I have returned the full subscription amount, and (thank god) J.J. is still speaking to me. I'll try to keep my explanation now similarly brief:

Nothing I stated about IMS (that obviously wasn't just my opinion) was false or incorrect. They had the software there at the Fest, they promised me a copy of it, and for over a month they did not deliver. I'd been waiting for ANYTHING new to run on the machine for exactly one year (I got the first unit at the Fest previous). To know it exists, and yet they won't give you a copy? And no explanation, just promises!? That was irritating me to no end. Combine that with the fact that I had held off the issue twice because I was told that it would be shipped right away, and you have a lethal situation. Well, my irritability with IMS obviously showed in my writing. (Can I get a vote on understantement of the year?) But I should state that a large percentage of my anger towards IMS is more because I know you can't run a business successfully with broken promises and endless delays, and I do want them to succeed. I knew I was going to get myself in trouble when I wrote the article, but I felt it might shake Paul into action. You see, it is the intensity with which I want to see these dream machines (and software) become reality that fuels my irascibility.

Okay, so I don't win any awards for brevity. Alright! Or professionalism. Gee, gimmie a break. Key...

OSK'er Back Issues Now Available through AniMajik Productions...

Issue #1	**"Premier Issue"**	**$5.00**
Issue #2	**"Playing Chess in C"**	**$3.00**
Issue #3	**"Hacker's Contest"**	**$3.00**
Issue #4	**"Accessing the New Year"**	**$3.00**
Issue #5	**"Where's the Beef?"**	**$3.00**

(Please include $1.50 S&H per Order)

Send to: **AniMajik Productions**
P.O. Box 8424
Universal City, Ca. 91608

This is Glenside CoCo Club

by Tony Podraza

About 6 weeks ago, a caller to the club BBS wanted to know what went on at the meetings, how many are in attendance, what the meeting times were, etc. I was tempted to respond then and there, but I let the message stand for a few days to see if anyone else would answer. The following is the response from Mr. Mike Warns who has been an avid supporter of the club for as long as I have known him, always ready to assist in any way he can. I might add, I didn't solicit his response, and at this time, he is unaware that his words are going further than he ever expected. But.....I digress. Mr. Warns.....

```
To:EDWARD STROH
From:MIKE WARNS
Subject:Club Meetings
```

The Glenside meetings can be very helpful and interesting, and the new administration is working hard to make them even more so. Because little changes are coming fairly often (at least, by the standards of a guy who manages to go to one meeting in three!) it's hard to precisely describe the "typical" meeting. However, this is a run-down of the last one (last Thursday):

The meeting was going by the time I got there (7:45). This is a big, new change because they didn't used to start anywhere close to on time. Tony Podraza (Senor el Presidente) was having everybody introduce themselves. This was followed by real, live Club Business (Tony seems to have found a copy of Robert's Rules of order--things were looser under Ed. It make things more businessike with more discipline and I have confidence in the future of the club.)

The next part was the most fun for me--RUMORS! Not just rumors and innuendo, but useful information, club & Coco news, stuff like that. There was a discussion of plans for the club to have a booth at the Atlanta Cocofest. There was then a vote whether the club members thought we should spend the money to do so.

We then had a bull session where anybody who had a question or problem could bring it up. This is where the club really shows its power--we have some of the leading lights of Coco-dom in this club. For instance, Eddie Kuns is a columnist for Rainbow as well as the Coco SIGOP on Delphi; Mike Knudsen wrote that really excellent MIDI program [Ultimuse 3] whose name has escaped me at 11:30 PM (although perhaps it has been demonstrated too often for the tastes of some members--our own fault is nobody else has volunteered to do a demo.)

There was also a demo of a new-ish word processor, a real MM1 in a real MM1 case. Roughly 2 dozen people were there. The meetings break up 9:30, 9:45ish and are followed by more meeting at a local restaurant.

All in all, Glenside's meetings can usually hold MY interest, and I'm just some MS-DOS user who got caught up in it when I had a TRS-80 Model One and I and I needed help using it. I do have a Coco 2, though, so the information can be useful.

In a very brief nutshell, that's our second Thursday night of the month. We try to get a lot of stuff into what seems to be a VERY brief period of time. While I don't feel that I've been very successful at it, I've tried to keep in mind that there are still RS-DOS users in the club and I would like to be able to balance the table on their side as far as the demonstrations go; the first part of the year of 1991 has been pretty heavily loaded toward OS-9 related items. Anyway, that is the goal of Sept-Dec, more RS-DOS support.

A Little Background

Glenside has been around since 1981-1982, meeting in various homes at first; not much different from any other CoCo user group. Few people remember the names of the charter members, the pioneers who spent $400 and $500 for the first 4K CoCos here at Glenside. I, myself, didn't know about the club until April 1984, and am not sure that I could faithfully name them, myself. My involvement with Glenside began after the first club president, Keith Geru, had served out his term and had been succeeded by Ed Hathaway. By that time, Glenside had found it's current home at the Glenside Public Library.

In March 1985, the club published it's first newsletter, at that time, yet unnamed. For over a year, it retained the identity of "GLENSIDE COLOR COMPUTER CLUB Newsletter 'XX" where 'XX' was the year of publication. With the advent of the CoCo3..Newsletter 'XX became known as the COCO 123, and has remained so until this day.

Largely the brainchild of Ed Hathaway, the newsletter has had several "faces" over the years. About eighteen months ago, the second editor of COCO 1 2 3, Mr. Bob Swoger, gave us yet another interesting newsletter format and direction of efforts, that of inter-club communications via newsletter exchanges. We are currently involved in that program to the tune of 18 clubs to whom we mail complementary copies of THE COCO 1 2 3.

The club has been supported by as few as one and as many as four BBS's at one time with the current number standing at three. They are as follows:

```
*********************************
*                               *
*    GLENSIDE COCORAMA BBS      *
*        708-587-9837           *
*      PinBall Haven BBS        *
*        708-428-8445           *
*           S&V BBS             *
*        708-352-0948           *
*                               *
*********************************
```

Let's hear about your User Group or Club? Send us Photos and a report on the doings of your club. Material may be sent to :

The OSK'er
User Groups
P.O. Box 8424
Universal City, Ca. 91608

From time to time, various new ideas are spawned in order to meet the ever-changing needs of the members. At one time, Glenside had a series of SIGs for people with special intrests, and those ran from digitizers to telecommunications to alternate operating systems and other computer systems. BUT, we were all still linked by the common bond of the CoCo in one form or another. As Mike Warns' message above indicates, even Model 1 and MS-DOS users can get something out of Glenside. Just what, I'm not sure...but I'm glad that Mike said it.

Glenside has been blessed with some very interesting people. Before the rest of the CoCo community ever heard of it, Glenside was treated to Ultimuse. KBCOM made its debut here. Hard drive interfacing with the Burke & Burke hard- and software saw early development in the Glenside area and Chris Burke has been a featured speaker at one of our meetings. The IBM compatible monochrome monitor adapter was conceived at one of our "meeting-after-the- meeting" get-togethers at the restaurant. Hawksoft graces Glenside frequently with their product demonstrations and new ideas. PegaSystems has recently shown us their new version of Xpres. Last, but certainly not least, Second City Software (now Kala Software) was born from two highly-valued Glenside Club members, David Barnes and Ed Hathaway.

I'm afraid to stop for fear of leaving someone out, but this is getting lenghthy. The bottom line is this; the organization, ANY organization, is only as interesting as the MEMBERSHIP makes it. The key to the success that Glenside has enjoyed over the past nine to ten years has been INVOLVEMENT! Our membership has been constantly involved in the self-help process. Indeed, that is the goal of Glenside as stated in our Bylaws.

Bylaws for the Glenside Color Computer Club

Objective: The Glenside Color Computer Club of Illinois is a not-for-profit computer club established to assist its members in the learning and the better understanding of Tandy's Color Computer.

I. Meetings:

A. Meetings shall be held on the second Thursday .....

and so forth.

I would encourage any member of any group to make themself available to those who are looking for help in planning and executing the functions of the group. Otherwise, it becomes a one-man-show and the one man will get tired out pretty quickly. In the ten months that I've held the reins of Glenside, I've had lots of help from many, many people, and that's as it should be, because it is not Tony's Color Computer Club, but GLENSIDE'S.

We are looking forward to a very interesting future. What with the advent of the 'coco IV's' from the non-Tandy manufacturers, the new users of the second-hand CoCo's as well the continued involvement by the "old timers", Glenside is a pretty exciting user group to be a member of. As Mike Warns has already stated, Glenside will continue to "hold my interest."

For further information regarding the GLENSIDE COLOR COMPUTER CLUB, you can contact:

```
Glenside Color Computer Club
C/O Tony Podraza
119 Adobe Circle
Carpentersville, IL  60110-1101
```

Or leave E-mail on any of the BBS's listed above.

Your articles and program submissions are always welcome. Articles should be sent in ASCII format. Call (818) 761-4135 for more information.

Call the

<PLAIN RAP>

BBS

(818) 894-0012

300/1200/2400 8-N-1

Supports: **OSK, OS9 LII, OS9 LI**
COCO 1,2 & 3 RSDOS

Over 1,100 Download Files

Part of the StG Network

Our 8th Year in Service to the BBS Community

SysOp: Jim Sutemeier

Sponsored by Sirius Software/Hardware

Official FHL Distributor

The MM1's are coming!

by Scott Griepentrog

Last time I wrote about the MM1, I wasn't able to cover any of the software that was to come with it because I hadn't received a copy. It was a few days after the issue went to print that the long awaited disks finally arrived. And even if I had delayed the issue even further, it wouldn't have helped. Because of problems caused by differences in the ROM bootstrap in my unit, It was almost three weeks later before I was able to boot the new software and *finally* get a look at the wonderful new windows. If my exasperation with IMS was a little obvious in my previous writings, please realize that it was just about 14 months between the time I received my first MM1 (with very little software) and when I was finally able to do something neat with it. Now I'm not really an impatient person, but the MM1 has been an awful long time in fulfilling the promises made by it's maker, and it has a few left to go.

The Distribution Disks

Along with the MM1 comes a set of six distribution disks in the high density 1.44 Meg format. These disks contain OS9 68000 V2.4 customized for the MM1, and a good number of utilities and free programs. The first disk has a bootstrap designed for a one meg system, as well as enough commands to get started. A utility for changing foreground/background colors, a gif to iff converter, a demo version of paint by Mike Haaland, a sound file player, a 'flicker' player, and Tetrix game, also by Mike. There are also some files for the demo programs, and the normal OS9 files. As new stuff comes out, the list of extras will grow. You'll probably see a few things by yours truly. The second disk has a bootstrap for a three meg system, and similar array of commands and demos. On the third disk is more stock OS9 commands, the C compiler, and some more programs such as a iff viewer, a gif viewer, smersh (gives shell command line editing using termcap), proff (a text formatter), oj (a script language), sterm (terminal program), and kermit. On the fourth disk is some standard OS9 C examples, the CGFX library (for the CoCo-like windows), DEFS files, LIB files, a file manager for PC format disks (for which a compatible driver has yet to be written), and source for kermit. The fifth disk has docs for uMacs, scripts for oj, and docs for proff and sterm. On the last disk is boot modules nicely organized into different directories for making a custom OS9Boot file.

Kevin's Windows

What should be the accepted standard for windowing software for OSK before long (although the lack of support by other manufacturers poses a threat to this) is, interestingly enough, not yet named officially. Some people mistakenly refer to it as MM1 Windows, but the fact is that although it has been developed mostly on an MM1 (because IMS has had the foresight to supply Kevin Darling with a unit), it has being designed to work with most any OSK machine.

Although certain features might require the VSC display generator chip used in the MM1, there is already a version of the same software working on the ST, which uses entirely different display circuitry. Kevin has even run several of the MM1 programs on the ST. The currently popular name for it is KWindows.

For anyone who is familiar with the windows on the CoCo, moving to KWindows is like going home. Almost all the escape codes work the same, although there is a new set of window type codes (because of the difference in display capabilities). The CLEAR key to switch windows has been changed to F10, with F9 to select the previous window and F1 through F8 to select TERM through F7 (provided you have them active). The use of the function keys in the windowing software prevents their use in application software currently, but Kevin says he's working on that. For programmers, it's a godsend. The compatibility with CoCo codes means that the amount of changes necessary to convert an OS9 program to OSK is reduced to a minimum. No need to use the termcap method in order to get the program up and running (although it's not a bad idea anyways). There is a noticable lack of documentation covering KWindows features, but this should not prevent anyone from using it or programming under it (provided you have a copy of the OS9 manual). Getting information on some of the newer features, however, will probably mean calling around (or catching Kevin on a break).

The Demo's

There are some really snazzy demo's that have come with this new software. The flicker program, my favorite, displays a repeating sequence of graphics images, creating an animation that is truly a must-see. The speed at which these run really demonstrates the capabilities of the machine, and running several at the same time shows off our multi-tasking capabilities nicely. There is also a sound player, a gif viewer, and another animation player. There is a demo paint program which looks really neat, but it is not useful for any real work. There may be a lot of software packaged with the machine, but you'll still have to shell out some bucks to get the programs you want.

The Ball is Rolling...

The MM1's are now shipping fairly regularly, and although there have been some delays on the I/O boards, these are said to be shipping as well. There is still some customer suport problems, because of the deludge of calls that Paul Ward has to deal with, and the limited amount of time he has to take care of them. I would hope to see someone hired to IMS to handle full time support in the near future, so that customers can reach someone a person instead of an answering machine. It can be hours to weeks sometimes to get a response from IMS. But, things are improving. Slowly. But then, what else is new in computers, right?

StG Net International

$49.95

StG Net Software/Login Pakg V3.0

Complete software to run your own BBS!
For OS9 Level II /Coco 3 (OSK Version soon!)
Extremely flexible!! ... Includes:

FREE Upgrade to V4.0!
Last chance to get V4 at V3 prices

- Run a powerful, multi-line BBS without losing the use of your computer! Run up to 8 lines.
- Hook up with an international Network, the StG Net.
- Net Account included for your system so you can "net" with other StG Systems imediately. Other Network interfaces coming soon.
- Complete E-Mail and Net-Mail Message System. Binary or Text files can be sent as private mail.
- Flexible Menu system allows you to create your own menus in ANSI or OS9 Graphics. Almost ANY OS9 program can be run from a menu (Std I/O). Sample Menus included, so you can go right on-line.
- DES (Data Encryption Standard) Password Protection
- System utilities include: Mail, News, Chat, TSmon, Login, Help, Netxfr, Option, Status... and many more...
- Help Utility included, gives you an On-line manual. Also includes printed insallation manual.
- Xmodem/YModem/Kermit Protocols for file transfer
- Includes FREE upgrades to Version 4.0 (Coming soon) All valid systems will receive upgrades via the net!
- Includes AniMajik's Games and Utilities Pak, made specially for the StG Net System.

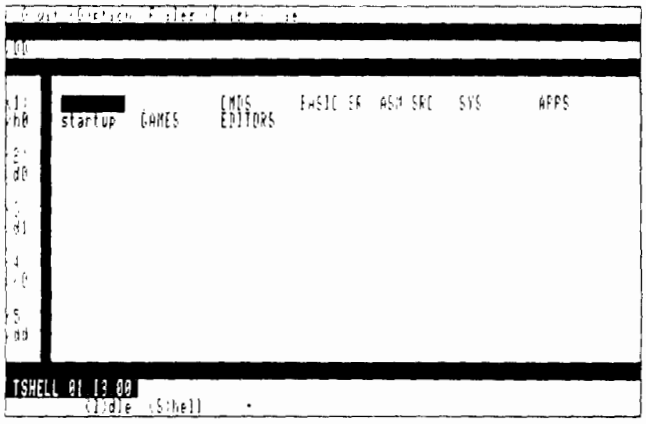
AS0011 (StG Net Login Pkg + CDI + IRQ Fix $49.95
(Please include $3.00 S&H)

Coco Tycoon - by AniMajik Productions

Create a "Monopoly" on your Coco3 (OS9 L2 Req)
1 to 4 Players Even play against the Coco...
Plays just like the Board Game...

(Reg. Price $19.95)
Special Intro Price $14.95
(Add $3.00 S&H)

(Take $3.00 off for Downloading via Modem/ S&H not Required!)



TSHELL - by Paul Pollock
(For your Coco 3, OS9 Level 2 System)

A Revolutionary New Program..."TShell" does most of what Multi-Vue does at many times the speed! TShell will run most programs with one keypress...and will use standard MV AIF files. Delete, Copy, Rename files all with one or two keypresses!

Many Utilities Included *** WINDINT and MV NOT Req!

AS0012 (Includes Disks and Printed Manual and accessories) (Please Include $3.00 S&H) $39.95
AS0012M (Save $13.00!) Download program Direct by terminal (From any StG Node, call for more info) Includes everything as above. Manual is in pre-formatted docfile ready for your printer. (No S&H Needed!) $29.95

AniMajik® PRODUCTIONS

(818) 761-4135 (Voice)
(213) 460-2968 (FAX)

MONSTEROUSLY GREAT DEALS...
AT MORE THAN HUMAN PRICES!!!

(Prices Subject to Change without Notice)
(Ca. Residents please add 6% sales tax)

Send Checks or M.O.'s

P.O. Box 8424
Universal City, Ca. 91602

Coming Soon!

* **DB9 -The Best Database for OS9!**
(Call or write for details and availability)

* **RECLAIM - The Disk Doctor**
"Reclaims" deleted files, fixes bitmap and sector problems on your floppy or hard drives.
(Call or write for details and availability)

* **MemMatch! - Coco Concentration...**
(512k and OS9 Level II Req.)

Call and Browse our Catalog at any of these BBSes
(At the LOGIN prompt, type "animajik")

(818) 761-4721 (MODEM)
(818) 772-8890 (MODEM)
(403) 329-6438 (MODEM)

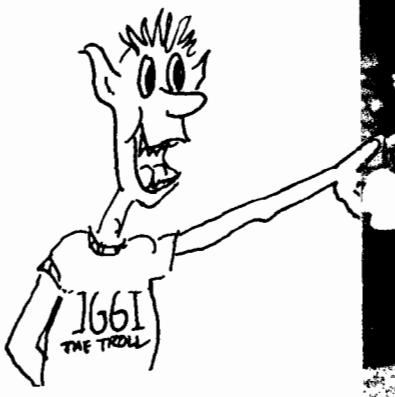
StG Net/West

BBS BREAK

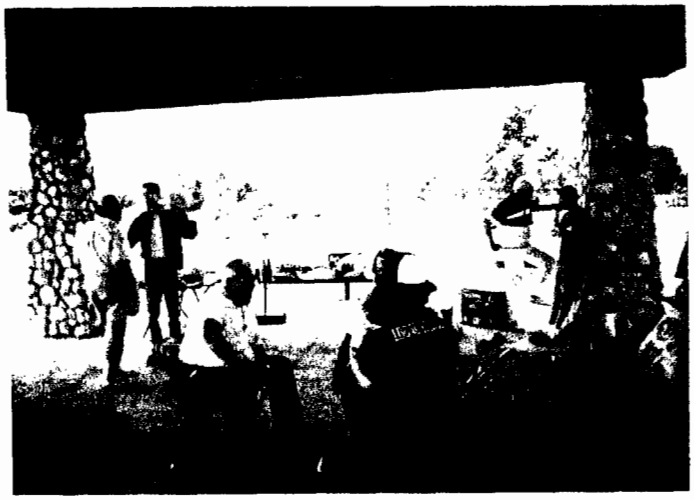
StG Net/West consists of five StG Net Boards in the Los Angeles area. These systems are "Zog's Cavern BBS", "PlainRap BBS", "Dune BBS", "Night Gallery BBS" and "Hall of Wisdom". Our First BBS-Break of the StG Net/West gave us all a chance to meet the "faces behind the Handles". Many who had chatted only via modem got a chance to chat in person. Pizzas and Iggi Cookies (baked by our busy baker Boobie) were consumed in mass quantities. Iggi the Troll has become a mascot of the StG Net and is currently the bartender of the ALT/TEN_FORWARD news area.

The Second Monthly StG/West BBS-Break was enjoyed by all with a delightful picnic at the new Balboa Park in Encino, Ca., Sunday July 21, 1991. Everyone chatted and asked questions about their favorite operating system, OS9, while hot dogs, soft drinks and cakes furnished by the crew were enjoyed by all.

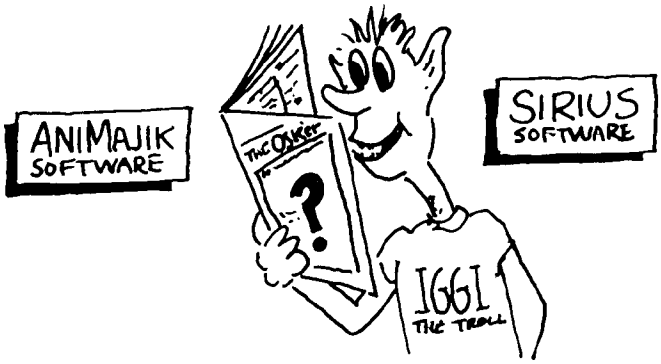
Since our last meeting, 2 new StG Boards have joined our group..."TSSERACT" and "HANGER"



From left to right...
Scotty (Scotty Grover), RCPilot (Dan Allen), Blarson (Bob Larson), ZOG (Alan Sheltra), Boobie (Mike Ortloff) PaulBell (Paul Pollock), Cosmo (Steve Secord), SHowman (Gary Cooper), Ion (Ion Michaelides), Maudib (Leonard Cassady), Jim (Jim Sutemeier), BlkRider (Anton Sipos), and Aloe (Allen Williams)



(Photos by Dan Allen)



AniMajik Productions and Sirius Software had their wares on display.

Boobie (Mike Ortloff), the Busy Baker, did the honors, cutting the first piece of his culinary masterpiece, "Green Iggi Cake".

Let's hear about your User Group or Club? Send us Photos and a report on the doings of your club. Material may be sent to :

The OSK'er
User Groups
P.O. Box 8424
Universal City, Ca. 91608-0424

Bottom L-R: Maudib (Leonard Cassady), Wayne (Wayne Campbell), Boobie (Mike Ortloff), RCPilot (Dan Allen)
Top L-R; PaulBell (Paul Pollock), ZOG (Alan Sheltra), Cosmo (Steve Secord), Flatline (Gene Turnbow), Blarson (Bob Larson)

StG Net/West - BBS Listing

System ID	System Name	Phone Number	SysOp
ZOG	Zog's Cavern BBS	(818) 761-4721	Alan Sheltra
PLAINRAP	PlainR Rap BBS	(818) 894-0012	Jim Sutemeier
DUNE	Dune BBS	(818) 992-4279	Leonard Cassady
GALLERY	Night Gallery	(818) 448-8529	John Powers
HALL	Hall of Wisdom	(818) 893-3839	Wayne Campbell

OSK ... is *OK!*

Moving forward

Interactive Media Systems, Inc. joins the software vendors below in welcoming OS-9/68000™ to our community. New windowing software from Kevin Darling, along with excellent support from HyperTech Software, has made many of your favorite programs available now, with the added power of fast computing and colorful graphics.

IMS, Inc. is shipping its MM/1™ computer, and other companies are offering OSK systems. We encourage you to find out about these systems and the new, exciting software soon available on them.Call StG and ask about windows utilities; call Color Systems and ask about label printing software. Call CoCoPro! and ask about moving around OSK directories; and call AniMajik about their keyboards, hard disk drives, and telecom software. Try Burke & Burke for their world-class utilities and programming tools; then telephone Brett at BeW Software about making the most out of the Bourne Shell and UUCP. Kala Software, JWT, and others, are creating your future.

We are your community

IMS, Inc. and all the vendors below are staffed by long-time Color Computer supporters. They are bringing you the next generation of computing. Their hard work is a symbol of their commitment to the most important component of the Color Computer. That's you.

Pick up the phone and call these companies. See how you can be a part of this new, exciting future. Thank you.

-- Interactive Media Systems, Inc.

Burke & Burke

Chris and Trisha Burke -- 206-432-1814
Dialog Box Manager • File System Repack

HyperTech Software

Mike Haaland -- 702-362-5346
Fontasee for the MM/1 • Paint • CGFX

AniMajik Productions

Alan Sheltra -- 818-761-4135
OS-9 and OSK software and hardware

BeW Software

Brett E. Wynkoop -- 212-567-7617
BeW Utilities • UUCP support • Send Fax coming soon

Color Systems

Zack Sessions -- 919-675-1706
For the MM/1: Variations on Solitaire • Sub Battle • Yah*zee

CoCoPro! Software

Dave Myers -- 313-481-DAVE
For the MM/1: Presto Partner, Data Windows; OSK: The Zapper

JWT Enterprises

Jordan W. Tsvetkoff
Nine Times Disk Magazine • Optimize™ Utility Set for OS-9/OSK

Kala Software

Ed Hathaway -- 919-333-1657
Call for catalog! Watch for UltiMuse/K for the MM/1!

StG Computing, Inc.

Scott Griepentrog -- 317-241-6401
StGNet • DB9 Multimedia Database • OSK and MM/1 Utilities

Interactive Media Systems, Inc.
202-232-4246

Call for brochures on these and other fine products

Introduction to BASIC09

By Eric Levinson

[Editor's note: this is part 3 in a series on BASIC09 programming]

In-depth parameter passing and complex data types

Now that you are familiar with basic parameter passing, I would like to elaborate on the basics. You know that the system keeps track of which variables are being "passed" by way of a stack, which can store data on a first in last out basis. In fact every programming language utilizes the stack in one way or another. You also know that the values themselves are not passed, rather the addresses or "pointers" to the data space where the variables are stored are passed. You know that variables passed by their location in memory is known as being passed by reference and that this is the default for BASIC09. You also know that you can pass a variable by value by attaching a null constant to the formal parameter list. This passing by value is really not desired especially with large data types.

Onward, James!

In last month's issue, I wrote two procedures to demonstrate parameter passing. The procedures were called ONE and TWO. In the simple demonstration I assigned X an initial value of 14, then passed it to procedure TWO. Procedure two then multiplied X by two and then exited. When procedure one resumed execution after procedure two terminated x had the value 28. Why? What would have procedure ONE printed if the fourth line was changed to:

```
RUN two(x+0)
[hint: pass by value rather than reference]
```

Passing by value is great when you want to manipulate variables in the called procedure without returning the manipulated variables back to the calling procedure. The line change in procedure one would have resulted in a 14 being printed regardless of what the called procedure did to the original value.

Okay, so much for review. Now we jump into the more interesting stuff, passing more than one data type, or passing complex data types.

Suppose you wanted to pass to procedure two the variables x, y, z, a$, b$ and c$. How would you do it? take a look at the following programs:

```
PROCEDURE one
DIM x:INTEGER
DIM y:REAL
DIM z:BOOLEAN
DIM a$:STRING[20]
DIM b$:STRING[30]
DIM c$:STRING[1]

x:=123
y:=1.2345
z:=FALSE
a$:="Is it a nice day "
b$:="in Sunny California?"
RUN two (x,y+.0,z,a$,b$+"",c$)

PRINT "Here is x: ";x
PRINT "Here is y: ";y
PRINT "Z is ";z
PRINT a$;b$
INPUT "Would you agree? (y or n)",c$
PRINT "The response to the yesno was: ";c$
PRINT "End Job."
END

PROCEDURE two

PARAM a:INTEGER
PARAM b:REAL
PARAM c:BOOLEAN
PARAM d$:STRING[20]
PARAM e$:STRING[30]
PARAM f$:STRING[1]

a:=5
b:=1.414
c:=TRUE
d$="Is it a terrible day "
e$="in Chicago, Illinois."
END
```

What do you think the output will be when you run procedure one? What does the +.0 and +"" do in the above procedure one example? How would the output differ if those arguments were taken out? Why did I use different variables in procedure two than in procedure one? Can I do that?

All of there questions can be answered. The output should read:

```
Here is x: 5
Here is y: 1.2345
Z is TRUE
Is it a terrible day in Sunny California?
Would you agree? (y or n)? [enter anything
you want here]
The response to yesno was: [whatever you
entered before]
```

The +.0 and the +"" are ways to tell BASIC09 that the variable is to be passed by VALUE, so procedure one passes to procedure two, address of X, 1.2345, address of Z, address of a$, "in Sunny California?", and the address of c$. Since the address of X is passed, any modifications to that memory area will effect X in procedure one. The PARAM a:INTEGER in procedure 2 tells BASIC09 to create a variable named A but in the same address that is passed to it, in this case X's from procedure one. Since 1.2345 is passed without the address of Y, procedure two cannot overwrite what is in Y's space, so the PARAM b:REAL in procedure two creates a new data variable space and stores 1.2345 there. The address of Z is passed to procedure two and its state is changed from FALSE (initialized in procedure one) to TRUE (set in procedure two).

The same thing is happening with the strings. The address of a$ is sent to procedure two and procedure two creates a variable d$ with the same address as a$ in procedure one, therefore any assignments made to d$ in procedure two will cause procedure one's a$ to reflect the change. However, b$ in procedure one, since the null variable "" is added to it, it gets passed by value. It is passed by value because once the +"" is postfixed, the expression b$+"" is looked at as a constant, since its value b$+"" can not be changed. Therefore the entire string "in Sunny California" is passed to e$ in procedure two, thus e$ is allocated separate memory for that variable and changes made to it are not passed back, causing the line "Is it a terrible day in Sunny California? " to be printed out.

You also notice that the passed variable names and the receiving variable names need not be the same, as a matter of fact, when a RUN is issued in a BASIC09 procedure, the variables of the called procedure are completely independent except for the passed variables. For instants, if you wrote a program called procedure one, and it used a variable called total_sales and a procedure two used a variable called total_sales, and they were not passed, they are two independent variables and one will not affect the other. Another words, all variables are LOCAL to their own procedure and ONLY their own procedure. This is called GLOBAL/LOCAL variables. A global variable is one that is passed to all the procedures to that the variable is accessable globally. A local variable is only accessable from within the SCOPE of its own procedure.

So much for value and reference passing. The rest of these articles will utilize passing by reference since it takes the least amount of memory.

Complex Data Types

So far you have learned four data types: INTEGER, REAL, BOOLEAN and STRING. These are the four building blocks that BASIC09 provide so you can design your own programs based upon these types. There is one other data type, the Complex data type. This data type is created by you, the programmer. Complex data types can consist of one or more of the four primary data types, in any order as long as field names are not used more than once. The TYPE statement is used for linking different data types together.

Suppose we wanted to write a phonebook program. We would want to set up three fields. Last name, First name and Phone number. One way this might be done is like this:

```
PROCEDURE phone

TYPE record=last:STRING[32];
first:STRING[32]; phone:STRING[10]
```

What we have done is created a new typename like INTEGER, called RECORD. We cannot access RECORD directly. Instead we can now define a variable called rec as type record:

```
DIM rec:record
```

Now our variable rec contains the entire record. It is 74 bytes total in length. We can access the three fields by entering a period (.) between the variable name and the field name.

```
rec.last := "Levinson"
rec.first := "Eric"
rec.phone := "7148316530"
```

The above shows how record assignment would be done. You cannot assign rec directly to anything, except of its own type. For instance, given the above example, if I enter in:

```
DIM rec2:record
```

I can assign rec2 to rec1 one of two ways:

```
rec2.last := rec.last
rec2.first := rec.first
rec2.phone := rec.phone
```

or:

```
rec2 := rec
```

Obviously the latter is easier. You can assign variables of the same type, or even different types as long as they are the same family, you can't assign a string to an integer unless you use the proper conversion function, or write a program to do the conversion for you. There will be more on records in later articles.

A record can be passed just like a single variable, you need only show the variable name (in this case rec or rec2 would work nicely.)

To pass it, simply issue a RUN,

```
RUN print_it (rec)
```

and rec will be passed to the called procedure the same way other data types were passed. Like the simple data types, you need to have a PARAM with the EXACT specs on it same as the calling procedure, so you will need a TYPE:

```
TYPE record=last:STRING[32];
first:STRING[32]; phone:STRING[10]
```

and then followed by a:

```
PARAM rec:record
```

in the called procedure. Your two procedures may look like this:

```
PROCEDURE phone

TYPE record=last:STRING[32];
first:STRING[32]; phone:STRING[10]
DIM rec:record
INPUT "First name: ",rec.first
INPUT "Last name : ",rec.last
INPUT "Phone number AAAPPPSSSS: ",rec.phone
RUN print_it (rec)
END

PROCEDURE print_it

TYPE record=last:STRING[32];
first:STRING[32]; phone:STRING[10]
PARAM rec:record
PRINT "First name: ",rec.first
PRINT "Last name : ",rec.last
PRINT "Phone number: ",rec.phone
END
```

When you run PHONE, it will ask you to enter First name, Last name and phone number. It will then pass the entire record to print_it, which in turn prints the record to your screen. The nice part about this type of parameter passing is that no matter how many fields you have defined in your TYPE statement, it only sends one address to the calling procedure which in effect allows your program to run more efficiently and will provide more speed, especially if this routine is executed many times, within a loop.

At any rate you should have an idea of what parameter passing and complex data types are about. My next article will focus on random access files, we will expand on this phonebook program and make it usable.

If you have questions or suggestions, feel free to write the OSKer magazine, or my address. Questions and answers will be published in the magazine.

You can reach me on Zog's Cavern BBS, Color Galaxy Milky Way (preferable) or write:

Eric Levinson

24415 Marquis Ct.

Laguna Hills, CA 92653

"You can pass whatever you want as long as it isn't out!"

Until next time, have fun and keep up the BASIC09!

Why not do your OS9/OSK Machine a Favor?

Subscribe to "The OSK'er",
News and Views in the world of OS9/68000 and 6809

Get the latest news about OS9/OSK and the systems that run it.

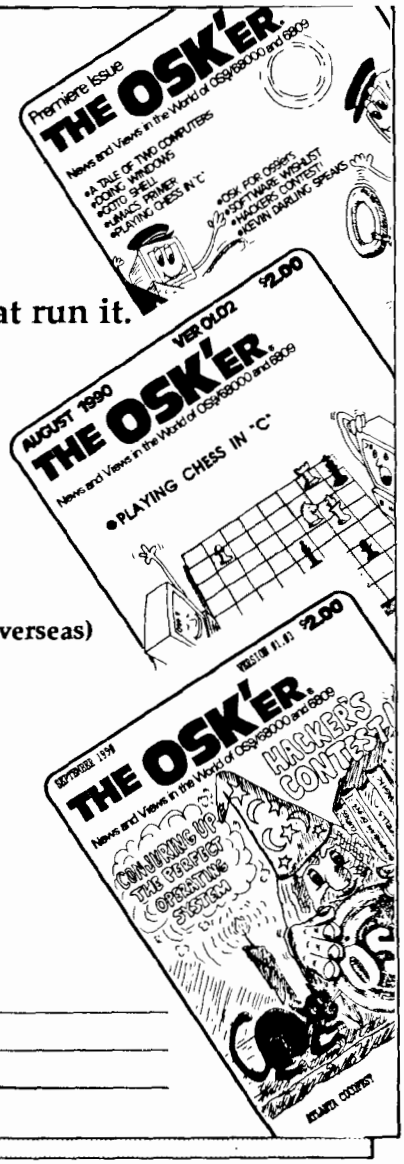
Reviews on the MM/1, SysIV, Sys030 and the TomCats
Columns on C, Basic09, and OSK Basic
Software Reviews
BBS and Club Listings
Festival Reports
Articles by well-known authors
...and Much much more!

For a 1 Year Subscription (12 Issues) Send $12.00 ($15.00 Canadian, $20.00 Overseas)

To: "The OSK'er"
P.O. Box 24285
Speedway, IN 46244

☐ Bill Me ☐ Payment Enclosed

Name: _____
Address: _____
City: _____ **State:** _____ **Zip** _____



Review: StG-NET

by Paul Pollock

[Editor's Note: with the exception of replacing "StG-BBS" with "StG-NET", and where marked with []'s, the entire text as submitted by Paul has been included unchanged. StG Net is the official name of the software being reviewed (previously called StG Login/BBS Package).]

Product Overview

'StG-NET' is a set of programs and utilities required to operate a Bulletin Board System. Written by Scott Griepentrog, this software comes complete with a re-written and improved TSMON, a substitute for LOGIN, and many support files - either re-writes of existing system utilities, or completely unique to this package.

Some of the support modules and message-base utilities have been written or re-written by Paul Jerkatis, and Alan Sheltra. Indeed, many of the cosmetic improvements are included with the standard package as the 'Animajik' pack (a set of programs by Alan Sheltra, included under marketing arrangement between Scott Griepentrog and Alan's company, 'Animajik').

This software runs under OS9 Level-2 at present. For purposes of this review, comments and discussion are appropriate for revision 3.0x of this package.

Market selling price for this package (at time of this review): $49.95. And this price includes free support, improvements; and a free upgrade to revision 4.0, when it becomes available.

Getting Down to Business

Firstly, I must preface with several important caveats.

1) I am an ex-sysop, so most of the discussion that follows will be appropriate to MY understanding of BBS operation.

   Attached to that is the fact that my experience was with PBBS (by Steve Roberson) which is a mixture of good and bad. It did not work via standard I/O, but did have many fine features. Throughout this review, and while I worked with and tested StG-NET, PBBS was the model I had to measure against.

2) StG-NET was delivered to me via several 'AR' and 'ARC' files from Zog's Cavern BBS (Alan Sheltra's BBS). Much trouble and time was spent setting up and diagnosing small problems with incomplete archives and program-files which were mis-matched in revision or capabilities.

   Some of these problems were solved, while others are still extent, at this time. This is not the fault of the package, as I got an ersatz copy of the package to review, rather than a complete 'startup' matched set.

   I think you'll all agree, that under these circumstances; any difficulties related to these events are not the fault of the software author, or I. So please adjust your reading accordingly.

3) My hardware is not standard Color Computer-3 fair. I have a Diskmaster, and the serial port configuration tables and other system related functions are much closer to OEM type OS9 Level-2 than the software included on the 'standard' Color Computer-3 variety.

   In truth, Color Computer-3 OS9 Level-2 is completely unique, not the least odd, is such items as the abbreviated baudrate code tables and port setup schemes we all take for granted, but other systems would find extremely unusual.

   These differences in port operations prevented the TSMON within the StG-NET package from properly operating AUTO-BAUD detection and other related functions. For instance, the baudrate table in Color Computer ACIAPAK has seven(7) legal values. My DMACIA contains fifteen(15) baudrates from 50-baud to 19200-baud. While the Setstat calls used inside the StG-NET TSMON would otherwise have worked, they would have had to send values not even present in the Color Computer-3 OS9 manual.

   For that, I have to apologize for any problems I ran into. The reader should be made aware that anyone using the StG-NET with the standard drivers that come with Color Computer OS9 Level-2, or driver software which has recently emerged into the public domain, intended for same; will work correctly as advertised.

4) After I was delivered the package from Zog's Cavern (initially Alan wanted me to work with the latest features and improvements), I now have to admit, that I think I'd have had a better picture of this package if I had been sent a standard package, without frills. I'd have had fewer difficulties with setup, and could then have enhanced the package at another time. Such is the folly of using hind-sight after the deed is done <grin>.

Now for the meat of the review. Initial impressions of the package have to lead to a favorable 'thumbs up'. While I was not able to examine some features and capabilities, I must be honest, and tell you all, that this package is extremely well thought out. Even without help files, or a instruction manual, this packaged could be setup and used with only a little guidance from present sysops. Because there is so much to examine, the following will be short discussions of major areas of features. Minor points (hopefully) will be touched on along the way.

However, usage of this software requires a firm understanding of OS9 file-structure, system rules, and SHELL usage, for installation and use. If you're in doubt, don't become an OS9 sysop.

Security

This package includes a global access data file called 'CIA'. This package is written to, and modified by LOGINS (a setup program), but CIA also has imbedded into it a secret code number for each system operator. This code is loaded and checked by every primary program module in the package, to ensure that the program(s) are being operated by the valid system, and that appropriate checks are still in place.

Additionally, the CIA also contains setup data which informs all operative package programs of salient system features like what directory is to be used for BBS commands, or where the SIG-menus are. CIA is the means used by the programmer to get this data passed from program to program, regardless of where they loaded into memory.

CIA has good and bad points. Good points include a high degree of system and file security. Indeed, Scott Griepentrog uses the DES encryption scheme to encode the imbedded system ID for each system operator. Each is unique, requiring several hours on an OSK machine to encode. Also important, is that CIA is used as a 'postbox' drop point for such features as the CHAT mode program and other tools. This allows the CHAT program to be used in tandem with other software, regardless of how many lines and ports are active (up to eight(8) can be accommodated simultaneously).

Bad news includes the fact that the reading scheme, used by each main program, to interrogate, gain access, and read the CIA (which must be done by most of the BBS programs) requires an enormous amount of computer time. On average, this delay is over 2 seconds of computer time (assuming the BBS is the only thing running); although on my system, with no netfiles or SIGS loaded, it seemed to be ready much faster (closer to 1/2 second on average). Indeed, dependant on hardware and system features, the BBSed program (the StG file editor used for message edits, etc) can not seem to get ready to edit a file in less that 5 seconds. My experience with these delays are with actual online time on several StG-NET's around the country, including the ROOT system in Speedway, Ohio.

[Editor's Note: the ROOT system, originally located in Speedway, Indiana, was lost during a recent move, and has been given a proper burial. A new primary system operating on an OSK machine is slated to become operational with the release of the next version.]

These delays are not to be confused with the file-load CRC. These delays have been checked and double-checked with other systems, and my own, with the file-load CRC DISABLED! While the average person might think these delays are nothing; when you use a BBS to leave numerous messages, work with your workspace, transfer files, or upload/download software; these long delays add up to major phone time. I consider this point, a major criticism, and needs to be dealt with.

Note: Scott Griepentrog has informed me that Version 4.0 of this software is a complete re-write of the BBS package, and has gotten away from CIA and other system delays altogether.

Menu's

Scott has done an outstanding job of providing for the desires of the sysop to customize the appearance of his BBS. He provides a menu driver that allows a person's menu to not only look the way he wants, but can also accommodate OS9 and ANSI graphics. Indeed, the graphics capabilities of each user is saved in a user-profile, and is remembered from logon to logon. This way, not only can a sysop setup menus that are clean and useful, but can also be razzled up with color, blinking or underlined characters, etc.

Criticisms are also in order. Firstly, MENU can only handle files which are fed to it as essentially 'top-down' written, to the screen. This is fine for general purposes, but many complex menus and pictures are often more efficient if only the absolutely necessary data is sent. The way MENU operates, in normal methods, a full-screen menu can take a huge amount of time to print.

Nextly, and is a personal view only, MENU can transmit ANSI graphics codes, but the StG-NET package has no tools for interpreting ANSI input. Half an ANSI is no ANSI in my book. Most terminals in OS9, which support ANSI, also transmit ANSI. This means that if you wish to edit a file, or send ANSI specific data to the system, it will be seen as ordinary ASCII text; with no action taken. While previous editions of BBSed could respond to ANSI codes, for message-file creation, latest editions no longer support it. If the system cannot respond to ANSI, it should delete the entire feature in outgoing data as well. It will confuse people that have ANSI capable terminals, and mislead folks logging in who don't use anything else (IBM, etc).

None of this takes away from the strengths of the overall scheme, but it should be attended to. And several programmer-sysops have moved at times, to their own methods to generate menus; for exactly the reasons stated.

My own feelings regarding graphics in general are as follows. I use OS9 graphics modes, quite often, when I logon to local BBS's. But if I had to operate non-local BBS's, I'd have to opt for standard TTY mode. OS9 and ANSI codes transmit a huge amount of data, which adds nothing to overall intelligence.

This extra data, while important for the system to handle things like color and format, can triple the time it takes to transmit a certain defined piece of intelligence. Indeed, some OS9 codes require 5 bytes of data, plus intelligence; while some ANSI codes contain more than 10 bytes, plus intelligence. The importance of this to folks that pay for their own phone time, can't be minimized.

Most StG menus are fed, in a manner that requires that escape sequences be sent for each actual character of data. This body of data, can become intolerable. On the other hand, these facts are a 'user beware' item. It's up to logged-on users to decide how much they wish to spend on phonebills, and how much time they wish to allow for printed data to be sent. But in general, my feeling is, support and service in OS9 or ANSI graphics, does not serve the bulk of users. The only people who win, are the owners of the phone company <grin>.

Upload/Download Support

A complete set of file transfer utilities and a very simple file-support menu set, is included in the package. Protocols supported are:ASCII Xon/Xoff, Xmodem-CheckSum, Xmodem-CRC (both standard and 1Kbyte), and KERMIT. Zmodem is being tested even as we speak, so watch for it in the future.

Most system operators have moved away from the supplied file-transfer menus, and are using the Animajik support programs. They are powerful and comprehensive.

NOTE:although they do take some considerable setup time, most [alternate support programs] require that directory tables initially are setup as edited program lines inside the program source, this requires experience with Basic09 as a programming environment.

Message-Base/Sig/Mail Support

Here again, while the package includes a set of SIG support menus, most sysops are now using the Animajik set for these functions. They too require some rework for directory tables (both SIGS and MAIL), but are otherwise very useful, friendly, and powerful. Even now they've continued to be improved, and are now almost CheckSum to commercial info-services.

Network Support

Scott Griepentrog and many other users and sysops, place CheckSum of value on StG-NET's capability to do complex networking with other BBS's in the net. This IS useful and constructive, in theory (I'll speak more about this later). The network can transmit any kind of data created on the system; data, text, messages, mailgrams, programs, info, even software revisions for the BBS.

The net-transfer program and support tools work as advertised. They are the only package in OS9 (that I am aware of) that allows sending packets and receiving packets in full real-time duplex simultaneously. This raises the net-transfer CheckSum dramatically, without increasing the system overhead over normal simplex block transfers.

StG-NET is not the only 'net-able' package, but it's protocol is at least as good as any other, and probably has some advantages over others. It not only supports its own file-constructs, but can (so far in a limited way) also deal with UUCP network protocols and structures.

NOTE: USENET support is said to be CheckSum, for those of you who have access to a USENET node.

My general thoughts on netable systems follow. Networks are the way of the future. That, for me, is both good and bad. They are good because it allows many folks to communicate over an enormous physical and electronic area. And it allows folks to transfer files to others without an enormous overhead in cost, at the individual level.

However, without prudent usage by each person who logs onto such a BBS, the costs of communication are not reduced. They are re-distributed to the system operator. It is our responsibility as users, to operate a network BBS in a prudent manner. If we use the network frivolously, we injure everyone else; by inference. When a BBS becomes too expensive to operate, sysops tend to disappear. People used-to get upset by my comments about networks, and I was strongly criticized for my obstructionist views. Since then, many network sysops have reviewed my comments, and have found them valid.

Most users are transferring their communications costs to the network, rather than spending their own money. And I'm not pointing fingers, I'm as guilty as everyone else. These costs are considerable, becoming at least as high as the maintenance cost of the phone line. Some sysops are complaining that daily networking would raise their monthly costs to $50-$100 per month over their normal usage. And is one reason for why the StG network normally NETs about twice a week, at the most. Amortize this over a years usage, and it's not hard see that sysops must supply large amounts of cash to such an enterprise. It's real money, and is being entirely enjoyed by phone-company stockholders, immensely.

But worst of all, netable BBS's tend to push out the local non-netable BBS's from the scene. BBS's are not on the rise. They are in the decline, for this and many other reasons. Indeed, many users tend to call networking BBS's just for the novelty, even when sending messages to friends, when there is a non-netable BBS which might serve both users equally well. This is not the fault of sysops. It's the users'. But I digress from the review.

Multitasking/Multiuser

StG-NET can support up to 8 connected users, simultaneously. Each can not only operate in a discreet way, but can link and communicate with others online via the CHAT facilities. Meanwhile, the host system can still be used while the BBS is operating, by the system operator.

And all ports operate via standard I/O. What this means is, the overhead is handled at the driver level, and has a minimum of system interference. This does not mean that everyone can fly at top speed. Folks running tests with multiple port BBS's have found that 2 ports at 2400 baud is just about all the Color Computer can manage (assuming no overhead increase from local system demands). For sysops wishing to have more than one port usage, it is highly recommended to downgrade port speed commensurate with specific port requirements.

The main feature of standard I/O (for those of you not familiar with this approach) is that any program run by StG will be able to make use of the serial port for all input and output, regardless of specific requirements. Write your program via standard methods, and StG can use it, externally <grin>. This makes program development, both from a BBS standpoint, and from an external host standpoint a dream come true. External users can call up this software, and program or write documents on your system. The files will be saved to system in conventional fashion, and will be waiting for you to deal with, when you arrive home.

This also means that a small OS9 Level-2 terminal could log on this software, get shell access and use the system remotely, as an example. Even screen graphics and pictures could conceivably be handled in this fashion. OS9 terminal codes are very rich, the potential for this kind of usage is enormous. Even overlays, and single-key graphic user interface menu's can be accommodated.

NOTE: Onscreen graphic single-key menus are used by several StG BBS's, and are known to have been first used in OS9 on the RCIS Color Galaxy Milky Way (a competing BBS system).

Closing Thoughts

I admit to having a little fun with this review. And I've been unscathing in my criticism. But I must complete this view with some basic concepts.

A piece of software of this complexity and sophistication, cannot (on balance) be spoken of badly. Scott Griepentrog has produced an extremely stable platform for BBS service, providing features and capabilities that have been pressed into service in ways never imagined by him. He overbuilt the package in many areas, and then sells it to users for much LESS than enough to compensate him for his time, energy and inventiveness. Indeed, I have seen BBS packages for IBM, which cost many times more, and provide less to the sysop.

This software, has evolved, and is still evolving. And I feel sure that it will become the standard by which all other OS9 BBS's (both 6809 and 680x0) will be measured. At this writing, I've become aware that Scott intends to port this package to OSK and to IBM platforms in the 4.0 versions. If he isn't considered a major player in power-networking after this work is completed; then he should proceed to the nearest sanitarium and retire.

But the strongest feature of this package, is not in the software. Scott himself is a major asset of the package. In support of this package, and his sysops; he is tireless, creative and generous. Unfailing is his attention span, ability to listen, sensitive to the feelings and needs of his sysops. And yet rakish and devil-may-care when dealing with new ideas and periods of invention. He is mature, and knows what he's good at, and where his capabilities leave off.

Over time (better than 2 years), I've watched as this fellow has worked to get his little network off the ground. He's proved to be very important in the great scheme of the Universe. He's an honest man, absolutely forthright. And there isn't enough of his kind, anywhere on Earth.

StG-NET V3.0, by StG Computers Inc.; Copyright (c) 1989,1990,1991, All Rights Reserved. Marketed through arrangement with ANIMAJIK (dba Alan Sheltra) (818) 761-4135 voice, (818) 761-4721 data; $49.95+shipping.

[The editor wishes to remind readers that the author is solely responsible for his comments.]

Color Computing

The bi-monthly magazine for Tandy Color Computer users

Attention all CoCoers...here's another CoCo publication to add to your collection! Color Computing is loaded with up to 35 pages of useful information for your Tandy Color Computer 1, 2, and 3!

Some of the features of each issue are...

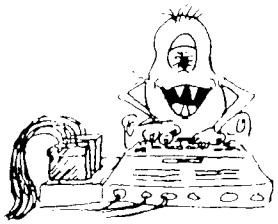
...Many interesting articles and useful programs
...Columns on OS-9, ML, telecommunications, etc.
...Hints & Tips

For a trial issue...

Send your name, full address, and $2.00 to:
COLOR COMPUTING, 65 OAK ROAD, CANTON, MA. 02021
For more information call (617) 828-7749

SUBSCRIPTION RATES: 1 year $13, 2 years $20

Advertise in the OSK'er Now!
Reasonable Rates!
Contact Alan Sheltra @ (818) 761-4135 for more information



AniMajik Productions
P.O. Box 8424
Universal City, Ca. 91602
(818) 761-4135 (Voice)
(818) 761-4721 (BBS)

AniMajik's Hardware Goodies

Cat #	Description	Price	S&H
#AH0035	Low Profile 82 Key XT/AT autoswitchable Keyboard	059.95	4.50
#AH0036	16- 41256 Chiips package (for Coco3 Only)	029.95	2.00
#AH0037	1 Meg (1 Meg X 8) SIMM Chips (each) MM/1 -TC70-TC9	054.00	1.00
#AH0038	4 Meg (4 Meg X 8) SIMM Chips (ech) For MM/1, TC70 or MACs	165.00	1.00
#AH0021	CDI (Carrier Detect Interface) (May be required for StG)	017.95	1.00
#AH0022	IRQ Fix (Fixes interupt problem on the Coco3)	00.4.95	N/R
AH0027-1	Teac 360 KB 5.25 DS/DD Drive	088.95	3.00
AH0027-2	Teac 1.2MB 5.25 DS/DD Drive	104.95	3.00
AH0027-3	Teac 1.44 MB 3.5 DS/DD Drive	082.95	3.00
AH0028-1	Quantum 52 Meg SCSI HD (17ms, 12 w/buffering) LPS Series	299.95	7.00
AH0028-2	Quantum 105 Meg SCSI HD (17ms, 12 w/buffering) LPS Series	3.99.95	7.00
AH0028-3	Quanttum 210 Meg SCSI HD (17 ms, 12ms w/buffering) Pro Series	695.95	7.00
AH0030	Baby AT "Pop-Top" Case - 200Watt P/S - Perfect for the Coco	6060.00	5.00

(Clifornia Residents pllease add 7.5% Sales Tax * Please allow 2-3 Weeks on some items)
(Prices Subject to Change Without Notice * Checks or M.O. Only Please)

Basic vs. C

[Editor's Note: The following is an exchange 'borrowed' from a BBS network detailing the same program written in both the Basic09 and C languages.]

Blink in Basic09

by Alan Sheltra

"One thing's for sure. Unless you're Einstein, you're not going to learn C as a first language. BASIC09, maybe, but considering the need for OS-9 to run it, probably not. Each language has it's strengths, but C is the only one that's portable, independent of the operating system you run it on. On the other hand, if you're writing COCO specific or OS-9 specific stuff anyway, who needs portability? Each language has its purpose, and its place."

My point exactly...! I just think that C, at least for most users and especially new users would find BASIC09 an easier language to learn and one that will fit most of their needs.

I would love to do a joint BASIC09 / C tutorial showing the differences between the two languages...

Since you've covered doing a clear screen in C, how about a doing a simple graphics tutorial showing different ways to do the same demo... You can do one in C, me in B09... I would like to show how to do this using the system, rather than using CLIB or in B09's case GFX2... here's an example showing how to do a "blink on/off" using a BASIC09 procedure...

Fire up BASIC09 and load "blink" then pack.

To run from std shell:

blink("on")

To run from Shell+ :

blink on

BLINK.B09

```
PROCEDURE blink
(* Turn Blink on or off
(* Works only on a Hardware Text Window

TYPE blink=cmd,code:BYTE
DIM blnk:blink
DIM stdout:INTEGER

stdout:=1

blnk.cmd:=$1F

(* Pass parameter "ON" or "on" to turn Blink on, "OFF" or "off"
(* to turn it back off

PARAM on_off:STRING[3]

DIM passed_param:STRING[3]

(* Trap Parameter ERROR
ON ERROR goto 10
passed_param:=on_off

IF passed_param="ON" or passed_param="on" THEN
blnk.code:=$24
ENDIF

IF passed_param="OFF" or passed_param="off" THEN
blnk.code:=$25
ENDIF

(* Send system codes to STDOUT
PUT #stdout,blnk

END

10 er:=ERR

IF er=56 THEN \(* Parameter Error
PRINT
PRINT " Usage : BLINK("ON") or BLINK("OFF")
PRINT
END
ENDIF
PRINT
PRINT " Error #";er;" encountered in Blink"
PRINT
END
```

PARAMETER PASSING

The PARAM statement in BASIC09 has a syntax identical to the DIM statement. The difference is that a DIM (DIMenctioned) variable allocates storage for that variable type. To quickly refresh your memory, BASIC09 has 5 basic atomic types:

TYPE	Values Allowable	Storage
BYTE	Whole numbers 0 to 255	1 Byte
INTEGER	Whole numbers -32768 to 32767	2 Bytes
REAL	FLoating Point +/- 1*10^38	5 Bytes
STRING	Charaters Letter, numbers etc	1 Byte per Char
BOOLEAN	True or False	1 Byte

Parameters do not allocate storage for the variable being passed, but instead describes to the "called" procedure what DATA type is being passed to it. There are 2 ways a parameter may be passed, (1) by Value or (2) Reference. Passing by VALUE usually means passing an expression, while by REFERENCE we would be passing a variable. In this example, we are passing by VALUE.

PARAM on_off:STRING[3]

> Tells BLINK procedure to expect a STRING of 3 characters.

In the BLINK procedure the PARAM is a STRING variable of 3 characters called "on_off". This is used to describe to BLINK what is expected. One important note: BASIC09 does NOT check data TYPE of a passed variable, only the SIZE, that is why a PARAM must match the called variable.

DIM passed_param:STRING[3]

> Allocates storage for DIMemtioned STRING "passed_param"

passed_param:=on_off

> Makes "passed_param" EQUAL to "on_off"

The DIMentioned variable "passed_param" is used in this procedure for error-checking. Since "passed_param" is identical in storage size and type to "on_off", a validation check can be done by making "passed_param" equal to "on_off". If the wrong value were passed, the error trap "ON ERROR GOTO 10" would catch it (more on ERROR trapping later).

The parameter being passed in the case of our BLINK procedure is a string of 3 characters which is either "on" or "off". This is checked using an IF/THEN statement to determine whether we wanted blinking on or off.

Any atomic type can be passed as a parameter, including complex TYPE statements which I will go into in the next BASIC TRAINING installment.

Parameter Pitfalls: One quick note on passing BYTE variables; since expressions containing numeric values are either INTEGER or REAL, passing these to a BYTE parameter ignore the low-order BYTE of the number and results are usually 0! To avoid this problem, pass your numeric values to an INTEGER or REAL in your PARAM statements!

Stay tuned for "Making use of 'TYPE' Statement...

Any questions so far???

If you'd like a copy of BASIC09 and C source code for the BLINK program you may request it thru : SysOp@ZOG, Zog's Cavern BBS (818) 761-4721 or send a blank disk with self-addressed mailer to: AniMajik Productions P.O. Box 8424 Universal City, Ca. 92608

Blink in C

by Leonard Cassady

"I'm curious about ARGC and ARGV... I know these are the argument count and argument vector, but can you explain their use a bit? I assume this is the only way to pass something to a C program..?"

Let's start at the beginning.....

All C programs MUST contain a function called "main", which is always the FIRST function executed. The compiler treats "main" like any other function. When the function "main" returns, the program is done. The operating system may supply arguments to "main". Arguments to "main" are not required, but provide a way to supply command-line strings to the program...other passing methods include file I/O, data structure tables, etc...

The first argument, usually called "argc", is of an "int" type-cast, (INTeger), and is the number of arguments in the command-line when the program is invoked. There is no way to anticipate the number of arguments before hand, but you do not need concern yourself with this, as it is determined at run-time, (program execution).

The string arguments, usually called "argv", are of a "char" type-cast and are arrays of string pointers to the command-line arguments. Each argument is separated by a space, (ASCII 32).

"Argv" is always passed as a string and must be implicitly converted if you plan to use it as an integer, float, etc...(the standard library contains functions for such conversions). IE: "argv[0]" would be the name of the program. "argv[1]" would be the first string after the name of the program.

(Note): You may access individual characters of "argv" by treating them as two-dimensioned arrays, (as shown in "blink.c").

"argv[1][0]"	the first character in "argv[1]"
"argv[1][1]"	second character in "argv[1]"
etc....	

You may call them whatever you want, but they must be these type-casts as the compiler expects them as such. They are called "argc" and "argv" by convention, but this is not etched in stone.

Now that we have a C "work-alike", (and I do use the term loosely), of "blink.b09", I think we should improve on it. I'll start out with simple changes, and please ask if anyone needs clarification on anything.

Actually, argv is a pointer to pointer to char. While this can in many cases be treated the same as pointer to array of char, they are not the same type and this is a cause of confusion to new C programmers. C does not allow passing of arrays to functions, and (unfortunately IMHO) silently converts array declarations in function headers to pointer declarations. I always declare main like this:

```
int main(argc, argv)
int argc;
char **argv;
```

main returns int, which should be the exit status but some C compilers get it wrong and ignore main's returned value. Always exit main with a call to exit() in portable C programs.

Ok guys...The following program is as close as I could come to Zog's 'PROCEDURE blink' program without using 'exotic' C code. I also tried to comment the code as clearly as possible so you can see what is happening.

The only parameter passing is done from the command line by supplying a option, (none or more than one will 'force' an error message and display usage.)

Notes: The option checker is of the crudest form and would be the first thing I would improve. It checks for a match of the second letter in the option passed for either an ('n','N'), or ('f','F').

The second improvement would be to use the "switch" command as this compiles to more compact binary code that the conditional "if" does.

Last would be the blink codes....I'd use a structure type and address the structure members, saving a byte or two...

What's next, Zog?....

BLINK.C

```
/* To compile, "ccl blink.c"  */

#include <stdio.h>                      /* format I/O header file       */
#include <lowio.h>                      /* system I/O header file       */

main(argc,argv)                         /* beginning of program         */
int argc;                               /* command line string counter */
char *argv[];                           /* command line string pointer */
{
  int path;                              /* output path                  */
  char on[2],off[2];                     /* blink code strings           */

  path = 0;                              /* set output path to stdout   */

  on[0] = 0x1f;                          /* set  'on' string to '1f24'  */
  on[1] = 0x24;

  off[0] = 0x1f;                         /* set 'off' string to '1f25'  */
  off[1] = 0x25;

  if(argc >2)                                  /* more than one option? */
    exit(_errmsg(errno," Too many options..."));  /* yes, exit w/msg */

  if( argv[1][1] == 'n') {                /* is the option 'on'?         */
    write(path,on,2);                     /* yes? write string to path */
    exit(0);                              /* and exit program            */
  }

  if( argv[1][1] == 'f') {                /* is the option 'off'?        */
    write(path,off,2);                    /* yes? write string to path */
    exit(0);                              /* and exit program            */
  }

  printf("\n Blink: Invalid option...\n");     /* no option matches...*/
  printf("\n Usage:  blink [opt]\n");          /* show usage.         */
  printf("  Opts - on\n         off\n");
}                                             /* end program         */
```

Second version of Blink.C

There are a few minor differences from the first version:

The option is now checked for an exact match, instead of just the second letter in the option. We've introduced the string function, "strcmp();", from the library. (and added the header file "string.h", where "strcmp();" is declared.)

Eliminated the "path" integer, and in it's place used the "STDIN" macro. Renamed the code strings to "code1" and "code2" from "on" and "off", and used "on" and "off" to hold the new test strings. Changed the cast-type of the code strings from "char" to "int".

Although there are a few more things we could do to improve the code, none of them will improve the execution speed, and the savings in size is around twelve bytes...(considering the operating system grabs 8kbytes to run it), the savings is moot.

Why did we change the cast-type of the "code" strings to an integer? C, by default, deals with everything as an integer, unless you implicitly declare it otherwise...

A "char" type takes one byte, but since as a "char" type we dimensioned the strings three bytes long, "on[0-2]", and an "int" type takes two bytes, we've save a total of two bytes for both strings and eliminated any need for type conversion by the compiler.

Why did we drop "path" and where does "STDIN" come from? The terminal I/O is handled by the STDIN path, (STDIN = 0, STDOUT = 1, STDERR = 2). These are declared in the "stdio.h" header file, so declaring "path" was unnecessary. Using "STDIN" is the same as "path", (both = 0).

I'd like to take a moment here and discuss macros. By convention, (meaning most other programs), marcos are capitalized so that when reading C source code, they're easy to spot. They also have the advantage of being declared, (#define), in only one place. To change all occurrences of the macro in the code, you only have to change it in one place.

Their disadvantage is that the pre-compiler,(C.PREP), actually substitutes the definition for the macro label in the code at compile-time, making the final code larger.

Functions, on the other hand, have only one copy in the code and may be very complex, using less memory and result in smaller final code.

The moral here is for simple operations, use macros....for complex operations, use functions. (The use of macros can get out of hand, so use moderation). Why did we add "code1" and "code2"? What was wrong with "on" and "off"?

"Code1" and "code2" were added so we would have a separate strings to place the blink codes in as it made more sense and increased the readability of the source to use "on" and "off" as the strings to test the command-line parameter. What is "strcmp" and what does it do?

I heard people say that C has little or no string handling routines. This simply isn't true....the problem lies with the understanding and use of pointers. I'll save the discussion of pointers for another time as the subject can cover chapters...

"Strcmp();" is one of many string handling functions already in the "clib.l" library. It "lexicographically" compares two strings and returns a value. If the strings are identical, the returned value is zero,(0).

For our use, all we need to know is that if we get a "0" by calling "strcmp();", the two strings match, and the rest of the conditional "if" statement between the brackets, "{ }" will be executed.

The "flow control logic" of the program dictates that if neither comparison equals "0", then print the usage and end the program. Since no is posting questions about the code, I'll assume that I'm clearly explaining it....(If that's not the case, then I've already lost everyone....please ask if you don't understand something.)

BLINK.C

```
/* To compile, "cc1 blink.c"                                          */
/* The header files must be in DEFS                                   */
/* "cstart.r" and "clib.l" must be in LIB                             */

#include <stdio.h>                      /* format I/O header file     */
#include <string.h>                     /* string functions header file */
#include <lowio.h>                      /* system I/O header file     */

main(argc,argv)                         /* beginning of program       */
int argc;                               /* command line count number  */
char *argv[];                           /* command line string pointer */
{
  int code1,code2;                      /* blink code strings         */
  char *on,*off;                        /* pointers to test strings   */

  on   = "on";                          /* set test strings           */
  off  = "off";

  code1 = "\x1f\x24";                   /* set code(n) strings        */
  code2 = "\x1f\x25";

  if(argc >2)                           /*   more than one option?    */
    exit(_errmsg(errno," Too many options..."));  /* yes, exit w/msg    */

  if(strcmp(argv[1],on) == 0){          /* is the option 'on'?        */
    write(STDIN,code1,2);               /* yes? write string to path  */
    exit(0);                            /* and exit program           */
  }

  if(strcmp(argv[1],off) == 0){         /* is the option 'off'?       */
    write(STDIN,code2,2);               /* yes? write string to path  */
    exit(0);                            /* and exit program           */
  }

  printf("\n Blink: Invalid option...\n");     /* no option matches...  */
  printf("\n Usage:  blink [opt]\n");          /* show usage.           */
  printf("  Opts - on\n         off\n");
}                                       /* end of program             */
```

M6809 Emulator

by Scott Griepentrog

The M6809 Emulator is a package by Bob Santy that allows 68000 OS9 to run 6809 OS9 modules. With it, all of your old CoCo OS9 software can still be used when you move any of the newer 68k machines. The package comes with a printed manual, and a disk containing the program itself, as well as enough source to let you customize the interface.

Installation

Getting it set up is a snap. The only file on the disk that you need is the M6809 module itself. It goes in your /dd/CMDS directory. Then create a /dd/M6809 directory, and place all your 6809 CMDS, DEFS, LIB, SYS, etc. directories and their contents in that directory. Then you need to delete certain original OS9 programs from the /dd/M6809/CMDS directory, such as procs, mdir, mfree, and makdir. The manual here needs to be updated to strongly urge you to delete the makdir command, which I found screws up your hard drive if it is using 512 byte sectors (check your dmode ssize). In fact, ANY program that accesses the hard drive directly, whether run under the emulator or not, should be avoided if the sector size is 512 bytes (this is the case on the MM1!). While in the emulator, any program that is not found in the M6809/CMDS directory will be run as a 68k program, so deleting the 6809 mdir (which doesn't work because memory is organized differently) will cause the 68k version to be used.

Getting used to it

To start the emulator, you can either specify a program to run or just enter 'm6809' to default to the 6809 shell. That's right, you can switch to the 6809 shell program running under the emulator! From there you can execute any 6809 program, just like you would back on the CoCo. For example, you can edit a C program, compile it, run it, all under the emulator from the 6809 shell. Not that you'd need to if you have a CoCo available, but for those instances when you don't and need to run something you can't do yet in 68k, it does come in handy.

How it works

To run a 6809 module, the emulator loads up the file from the M6809/CMDS directory, and simulates in software the 6809 instruction set. All OS9 calls are passed to OSK by carefully matching the compatible calls. Certain extra interpretation is given to certain calls, such as open. Whenever an open is called on /DD/... it is passed to OSK as /DD/M6809/... so that the 6809 version of files are used. The fork call causes a check for the 6809 version of the named program, which if found a copy of M6809 is forked to run it. If not, a 68k version will be executed if found.

Also interpreted is the write call when it outputs to the screen. Normally, the codes will just be written out unchanged, but if the tmode tabc is set to zero, the OS9 windows escape codes are translated based upon the currently set TERM variable. This way most any terminal can be used, even when a program is moving the cursor around. The termcap emulation is limited to text controls, of course. Programs doing graphics or fiddling with windows will not work. If you happen to be using KWindows though, most of these codes are emulated under that software, and such programs will usually work correctly.

M6809 comes with the necessary modules and source to allow customization of the interface to OSK. There are four .C's on the disk in which you can add support for OS9 calls, setstats, getstats, and screen handling. A makefile is included to recompile the source with the emulator object into a new M6809 module. Although not recommended for persons without knowledge of assembly or C, this is an outstanding feature.

Speed Issues

Because it is a software emulator, it takes a good number of 68000 cycles (from 30 to 300) to emulate each 6809 instruction. Although the faster speed of a 68000 helps to offset this, it doesn't always make the program run faster on a CoCo. It depends on what the program does.

For example, take a simple 6809 utility that does nothing more than scan every sector on the hard drive to ensure readability. That kind of program is actually fairly short and simple, and although it's doing a heck of a lot of disk i/o, the emulator can run it almost as fast as it would have run on the CoCo. Of course, the faster disk speed on the 68000 will actually make the program quicker. However, the same program with an added routine that searches each sector for a particular pattern will be considerably slower, because it is doing a lot of processing inside the program.

The difference in emulated speed between one program and the next depends on how much processing the program is doing. For example, the following program was compiled using M6809 running the C compiler in one minute, twelve seconds:

```
#include <stdio.h>
main()
{
  printf("Testing 123\n");
}
```

This was operating on a MM1 with a very fast hard drive, and using my own CC executive (which passes c.prep through c.opt on pipes). One minute doesn't seem like much to wait, but that can be as much as a half hour to compile a much larger program. On a CoCo, with the C compiler modules already in memory, the same compile used to take less than ten seconds.

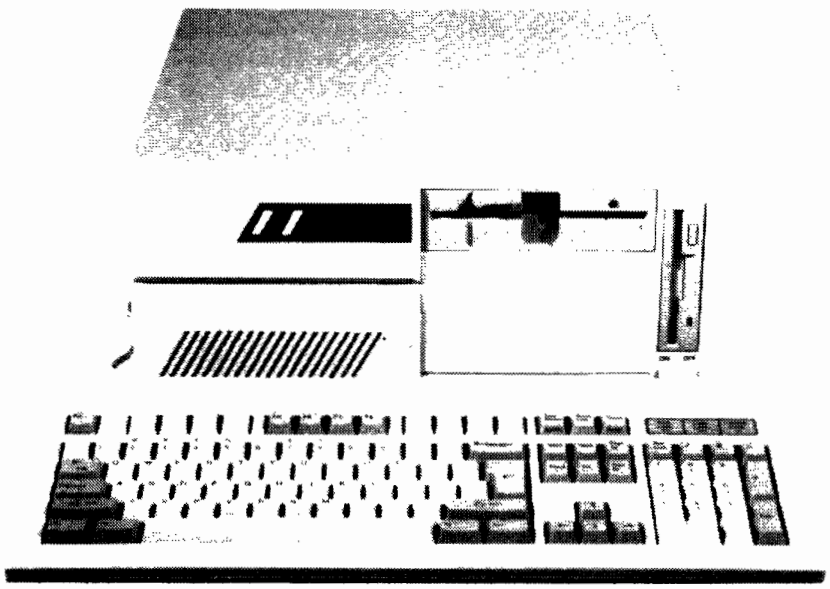
Final Thoughts

The M6809 Emulator is a very useful program for anyone who is moving to the 68k world from 6809. It's well written, works extremely well, and can be customized. Although a tad pricey, it's a valuable tool that I have found very handy, and recommend it highly.

The M6809 Emulator is listed at 99.95, and is available through Delmar Co, (302) 378-2555.

SYSTEM IV

The 68000 Computer serving customers here and abroad



THE BEST OF ALL WORLDS

- OS-9 Power - outperforms other machines in its' price class
- A Multi-User Multi-Tasking System
- Versatile - runs OS-9/68000, STARDOS, REXDOX, MS-DOS and other Operating Systems
- Flexible - tailor to your requirements
- Expandable using readily available Low-cost Cards
- Optional Plug-in Board for MS-DOS
- Optional Emulator/Interpreter for OS-9/6809 Software
- Ideal low-cost development platform
- Prices start at $999.00

For Kits and Assembled Boards call Peripheral Technology at 404-984-0742

OS9/68000 SOFTWARE

M6809 - 6809 Emulator/Interpretor $99.95
M6809 emulates the user mode OS9 Level II binary object modules under OS9/68000.

QUICK ED - Screen Editor and Text Formatter $275.00
A high quality documentation tool and program editor ideally suited to laser printer users. Uses function and cursor keys on any terminal, configurable per user. Micro-Justifies mixed proportional text. Automatic table of contents generation and user-definable macros and commands. Handles an unlimited number of fonts. Drives any printer. Ideal for multi-user systems. Available on a 30-day trial.

FLEXELINT V4.00 - C Source Code Checker $495.00
Flexelint finds quirks, idiosyncracies, glitches and bugs in C programs. 60 options control checking by symbol name or error number. Checks include intermodule inconsistencies, definitions and use of variables, structures, unions and arrays, indentation, case fall-through, type conversion, printf and scanf format string inconsistencies and suspicious semi-colons. A must for all serious C programmers.

DISASM_OS-9 - OS-9/68K Disassembler $250.00
This high-speed, three-pass 68000 disassembler can also handle the 68010 and 68020. It intelligently decodes module headers and produces symbol information that can be repeatedly edited and passed through the disassembler allowing iterative disassembly. The system libraries are read to supply symbols.

WINDOWS - C Source Code Windowing Library $250.00
This C source code library package supports multiple overlapping windows displayed on one character-based terminal screen. It supports window headers and footers, and pop-up windows. Windows may be moved, panned, written to while off-screen, etc..

PROFILE - User State Program Profiler $270.00
Designed to profile user-state programs. Profile effectively samples a traced execution building statiscal information as it goes. It reads symbol table modules to give a function-by-function account of the time spent during execution. The user may "zoom-in" on a function to find a smaller range of addresses where time is being spent.

delmar co

Middletown Shopping Center - PO Box 78 - Middletown, DE 19709
302-378-2555 FAX 302-378-2556