

THE NEW COLOR COMPUTER

by Bruce Warner

By the time this article hits the street, I'll have seen the new Color Computer. On July 30th, 1986, a press conference was hosted in New York City. The conference introduced four new MS-DOS computers and the new Tandy Color Computer.

This machine supports advanced graphics, up to 640 x 192 pixels in FOUR colors. The majority of existing hardware is compatible, including Deluxe RS-232 Paks and disk controllers, but much of the hardware may be undesired with the improved features of this machine. Incompatibilities will come from hardware and software that has violated specific conventions of programming standards both under

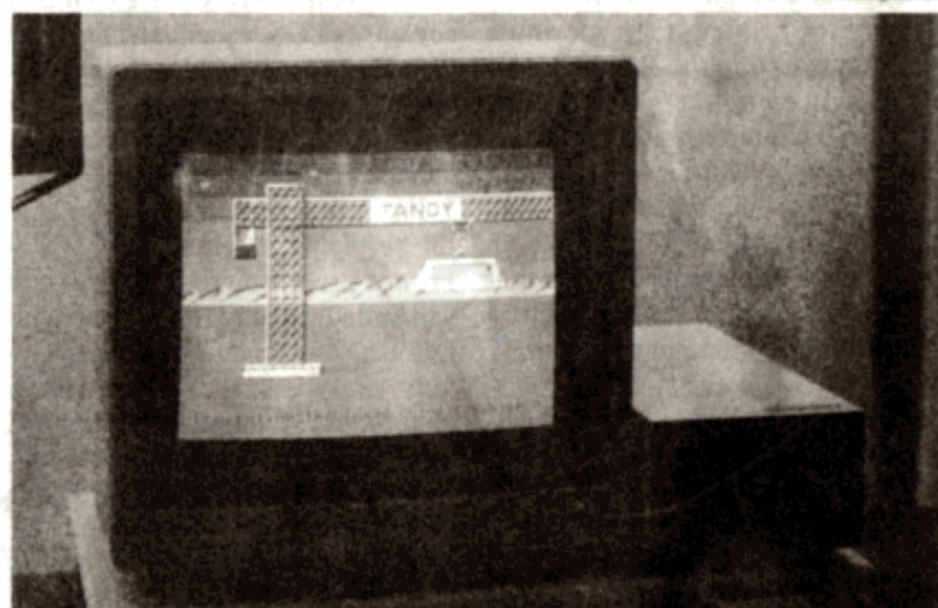
OS-9 and RS-DOS. Most of the programs are the ones that run in the "96K" mode, or those that make direct calls to hardware devices rather than using device descriptors and drivers. CoCo Max is among the programs that will not run, due to the 96K addressing used by CoCo Max. Most of the Tandy hardware will work without modification, since they have followed their own programming rules. An exception to the Tandy rule is the Multi-Pak interface under OS-9 Level II. A \$6 modification is required.



From the outside, the new Tandy Color Computer looks a lot like its forerunner, the Color Computer 2, but the insides make all the difference in the world.

The Software Compatibility Story

If you're using Telewriter under RS-DOS, you'll be surprised that it is one of the few programs that won't have compatibility problems. Telewriter-64 works because of its implementation of the 64K CoCo. Although the original version used undocumented calls to the ROM, which resulted in incompatibility of the software, the current version works because they didn't try to make Telewriter-64 work using 96K. This is due to the addressing mode of the GIMI chip. This is the chip that gives you all the power to



There's blue sky (four shades), a red crane, a black, white and red CoCo 3, green hills, orange foreground and black text. Altogether, 16 colors are available with higher resolution than the old CoCo and more programming memory than the old graphics modes.

address the extra memory. To make a "96K" program run, the GIMI chip must be set to 16K internal and 16K external ROM before performing any banking between the RAM/ROM and all RAM mode using the usual SAM addressing. The easiest way around this is to go to OS-9 and let the new Level II OS-9 system handle it all for you. This will also insure that your software runs first time out on any new Tandy OS-9 machine. RS-DOS also causes some problems with addressing memory that is reserved for the GIMI and SAM chips.

Tandy has listened to us, so you'll see more than you may have expected in a Color Computer, with memory starting at 128K and working its way up to 512K (in one step) and a variety of features. You'll even see graphics that rival the outhouse and game companies. We'll be able

to see this machine yourself in your local Radio Shack in a couple of weeks.

So what's so new and so great about a 6809 computer? Well, you may decide that you don't even need a 68xxx machine after seeing this one. Mostly because of the things that have made the OS-9 community so great to begin with, but with the added features of this machine it's a real bargain.

Besides the memory, you'll be glad to know that BASIC is still BASIC for those that want to stay that way. What makes BASIC better is that Microware (that's right, the OS-9 people) were contracted to do the patch for Enhanced Color BASIC. This includes 21 new commands aimed at making the best use of the hardware in this machine. This includes the new 80-character

Continued on Page 14

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 18
PROSPECT, KY

86/09/13

OS-9 Users Group
Attn: Membership
9743 University Ave., Ste. 330
Des Moines, IA 50322

Address Correction Requested

OS-9 AT NCC

(Photos and story — Pages 10 and 11)

Richard Don, owner of Gimix, Inc., shows off the Gimix that fits in a PC case. The use of the PC power supply and case offers the user the option of an exceptionally low-priced casing for their single-board computer.





PUTTING IT TOGETHER

The building of this month's MOTD

by Bruce N. Warner, Editor



September, 1986

Vol. 2, No. 7

The building of this month's MOTD may have been a little harder than usual. We'll start with the stupid complaints by people that have proclaimed themselves as OS-9 experts and go on to the daily routine of helping a friend whose son wants to join the Navy.

What in the world do these things have in common? Simple. They are two of the many things that occupy my world, from the few petty folks in the world of OS-9 (that take up most of your time) to the people that really have a problem and want a little shove in the right direction.

Puckett Jumps on the Soap Box

Dale Puckett has an excellent article this month on OS-9 critics. They remind me of Ralph Nader. Though Mr. Nader has spent a lot of time making us aware of how unsafe our cars and other items are, I sure am glad that I don't have to wait for the car he designs. I'm also glad I'm allowed to take some responsibility and not have to pay the extra \$100,000 it would take to put all the extra features Mr. Nader would want as standard equipment on our cars.

Dale hits the nail on the head when he tells us that "Those who can, do. Those who can't, criticize." If you're a critic, try being a doer. It doesn't take that much more energy, and it sure gets us into a healthier mood.

A Week in Vegas

Trying to cover everything in the world of OS-9 isn't easy. I just spent every penny I made editing the MOTD this year by going to one show. Not all of the shows are paid for by the UG, and the editor's post is the only paid position. The pay is less than adequate for what it takes, and I could make more spending the same amount of time writing for our local newspaper. The point is that I could be doing something a lot more profitable and much easier with my time. Like so many of the UG officers, I believe that OS-9 is worth the effort.

In case you're wondering why I refer to going to Las Vegas as an expense, I hit on the fact that I don't gamble and I don't drink, so the shows were nice, but the location isn't my idea of a good time. I really would have better enjoyed a week at a riding academy up north in June. The desert heat isn't my idea of great weather.

Coming up This Fall

If you're wondering about RAINBOWfest, you'll be pleased to know that we're all very proud to announce that the publishers of the RAINBOW are planning an expanded coverage of OS-9 seminars. Some of these will be because of the new Color Computer (see the article in this issue), others are because of an increased interest in OS-9.

I'm very proud to announce that I'll be giving a seminar on "OS-9 From a Users Standpoint." This talk will be dedicated to finding out what OS-9 means to those people who only want to use the software available on their computer. I, for one, rarely do any programming under OS-9. Instead, I purchase programs that are already written under our favorite operating system and learn how to use them and make the system work better for us all. I also get software and hardware from various companies and write the documentation for making it run under OS-9. This development of hardware and software documentation is one of my largest uses of OS-9, and it's greatest visibility is in the Color Computer OS-9 community.

We Don't Always Get to Talk

There are some people that think the UG officers talk on a daily basis. There are times that we don't even get to talk on a weekly basis. I haven't spoken to Brian Lantz but once in the last month. I do get to talk with Dale Puckett at least once a week, but Dale lives and works in the local area, and phone calls are cheap for us.

What's all this leading up to? Well, don't be surprised if you get different stories from different officers in the same day. We may get our heads together before the next time you give us a call, but at any given moment, we are just as likely as you to be operating on different frequencies. Add to that, there is always more than one way to look at and solve a problem. At one RAINBOWfest, Dale Puckett and I were answering several questions for people. Dale gave one set of answers and I gave another. While we were both right, there were times that Dale's answers were better and times they were worse than mine. They were all right, mind you, only looked at from a different person with a different point of view.

I keep remembering this incident when someone asks me what is the

best word processor for their OS-9 computer. I tell them I can't answer that, but I use DynaStar.

The next question is always, "Then why do you use DynaStar if it's not the best?" Simple. It is the best — for me! What's best for you could be StyloGraph or Last Word or DeskMate. I don't know what you're going to do with it or how you're going to make it work for you. I only know what's right for me. So, is my answer wrong? No. My answer is right under the circumstances I've chosen to consider to answer it. I use DynaStar because it's comfortable and it handles about 98 percent of my word processing needs. What you select is your option. What I select is mine. With the new Color Computer, we may find some really great word processors hitting the OS-9 community.

The point is that we can only answer your questions from our own vantage point.

Never Become an Expert

I have a friend who says an "EX" is a has-been and a "SPERT" is a drip under pressure, so what's that make an EX-SPERT? There's a much better reason for not becoming an expert. There is always someone out there that wants to rip your head off. As soon as your name becomes known, you're either a target or a source of information. It's kind of like being the fastest gun in the West. There's always someone that wants to shoot you out of the fastest seat.

I know of people that quote Steve Bjork, Dale Puckett and Brian Lantz for all sorts of things. One told me that Dale Puckett had been invited to the Tandy press conference in New York. I was glad to hear it, but a bit surprised. Dale and I had just spoken the day before and he had no plans or invitation to attend the conference. I, on the other hand, was planning a one-day trip to New York for the event.

The next day, Dale showed up at the monthly Saturday meeting of the Northern Virginia Color Computer Club, much to my surprise. What happened next was a little embarrassing for the member that told me Dale was going. Dale hadn't spoken with him or the people at Tandy. There was an assumption made because of Dale's reputation in the field and a desire to be asso-

Continued on Page 20

MOTD is published monthly as the official newsletter of the National OS-9 Users Group, 9743 University Ave., Suite 330, Des Moines, IA 50322

Editor

Bruce Warner
OS-9 Users Group
ATTN: Editor
9743 University Ave., Suite 330
Des Moines, IA 50322

President, National OS-9 Users Group

Brian Lantz

Vice President

Bill Turner

Treasurer

Steve Odneal

Secretary

Dave Gibson

Director-at-Large

Dale Puckett

Software Librarian

Dave Kaleita

Finance Committee

George Dorner — Chairman

Membership Committee

Joe Dubuc — Chairman

Public Relations Committee

Bob Rosen — Chairman

Software Exchange Committee

Carl Kreider — Chairman

Software Reviews Committee

Marcus Johnson — Chairman

Subscriptions to the MOTD are free to members of the OS-9 Users Group. Non-members may subscribe by writing to the chairman of the membership committee and paying an annual fee of \$20. Send to: OS-9 Users Group, ATTN: Membership, 9743 University Ave., Suite 330, Des Moines, IA 50322.

MOTD is produced and distributed by Falsoft, Inc., publishers of THE RAINBOW, the official magazine of the National OS-9 Users Group.

OS-9 is a registered trademark of Microware. There is no connection between Microware and The National OS-9 Users Group, nor is there any connection between Falsoft and The National OS-9 Users Group.

MOTD Advertising
c/o Bruce N. Warner
14503 Fullerton Road
Dale City, VA 22193-2034
Phone (703) 670-4962
(Include check payable to the OS-9 Users Group)



NEW LOW COST 68020 BASED MULTI-USER SUPER-MICRO

The **QT 20x™** is an extremely cost effective multi-user, multi-tasking micro that will support 2 to 30 users. RAM is expandable from 512K to 14.5 Megabytes; ROM is expandable to 256K. A number of useful software packages are included in its low cost. The **QT 20x™** can be purchased as a complete system with floppy and hard drives; in a kit form, or the motherboard can be purchased separately. Other hardware configurations can be customized to meet your needs. Listed below are the principal hardware and software features:

HARDWARE Features:

- 32 Bit MC68020 Processor - 12.5 MHz
- 512 Kbytes of 32-bit wide, high speed RAM
- 2 RS-232 Serial I/O Ports
- 1 Parallel I/O Port
- Built-In SCSI/SASI Interface With DMA
- Floppy controller with DMA
- 32 Bit Expansion Buss
- Real Time Clock w/Battery Backup
- DIP Switches to Select Boot Up Baud Rates and Boot Device

HARDWARE Options:

- 2 Megabyte RAM / 4 Serial Port Expansion (a maximum of 7 boards may be used)
- MC68881 Floating-point Coprocessor
- Memory Protection **
- Graphics Board **
- 4 Serial Port Expansion Board **

** (Available 4th qtr)

SOFTWARE included:

- OS9/68K - Operating System
- Basic09 - Programming Language
- Stylo - Word Processing
- Mail Merge
- Spelling Checker
- Dynacalc - Spreadsheet
- fbu - Hard Disk to Floppy Backup ***
- QCom - Communications with XMODEM and text file transfer ***

Other Software Available:

- 'C'
- Pascal
- PC Utility (Read/Write/Format IBM Disks)
- CRASMB Cross Assembler
- Sculptor Plus - 4th Generation P.L. (Data Base Management)
- Fortran 77 (Available 4th qtr)

Note: *** QCom & fbu are included FREE when a complete system is purchased.

LIMITED TIME SPECIAL

Complete system with 20
Meg Drive only **\$4495!**

2.5 Meg RAM, 6 Serial Ports 1 Parallel port, RTC.

1976  1986
FRANK HOGG
LABORATORY
770 James St., Syracuse, NY 13203 Telex 646740
315/474-7856

OS-9 U G SOFTWARE COMMENTARY

by Bert Schneider

Wow! 39,168 sectors — I never thought I'd see that many sectors on my Color Computer. I finally got the hard drive up and running on my system thanks to people like Ron Schmidt of L.R. Tech! The only problem I have now is that I have been unable to get OS-9 Version 2.0 to boot up on my system using the hard drive (I have all the upgrades for 2.0). Anyhow, I can at least boot up 1.1 without any floppy drives.

NEW UPDATES! Some of the User Group Library Disks have been updated to correct bugs or add more efficient code, or just to add some new routines.

Dave Kaleita has improved his documentation generation program DOGEN2 (a program used by software authors who want to contribute to the OS-9 UG Library). Dave has included on Disk #0 a new version called DOGEN3. He states that it cannot be packed on a CoCo, but I did it on mine without problems (I am also using Wordpak which frees up a great deal of memory). He also includes a new utility called LOWUP that converts all text to uppercase only. His DOGEN3 program requires all uppercase for data input.

Disks #6 and #7 have been fixed. The adventure program had a few bugs that would really irritate an avid adventurer. After spending hours down in Colossal Cave, you might suddenly find your adventure curtailed by a FATAL ERROR \$38 or other such messages. Those have been corrected on both the source code disk and the object code disk. Other disks in the library have been updated too. Since I have not covered those yet, I will just treat the new updated versions as the standard baseline disk for commentary purposes.

Disk #10, "Math and Electronics," fits right up my alley since I am an electrical engineer. All of the code on this disk is written in Basic09 and was put together by George Dorner (another OS-9 UG superman). I only wish I had access to some of these routines when I was doing some of my undergraduate work and even my graduate work. Some basic understanding of mathematics and electronics is probably required to use these routines. I don't think the average computer owner/user will have a need for these routines. However, the serious electronics hobbyist, student, or designer will find these programs most beneficial.

The students can learn more easily from comparing their answers to those found from the programs. Usually students are limited by only a few examples in text books. These programs enable the student to go beyond just the basic examples. As for the electronics hobbyist or designer, no one really sits down and designs by hand anymore. Sure, the designer has to come up with the idea, but the computer can do all of the tedious job of number crunching (which if done by hand is prone to error). This disk demonstrates that there is more to computing than just system utilities or file processing.

NAME: rc

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$1CB

DESCRIPTION: This routine is a simple Basic09 program that prints out parameters of an RC circuit (resistor-capacitor) as the resistor heats up.

USE: run rc <cr> or rc <cr>

You must modify the initial conditions inside the program itself for each case. Therefore, it would not be wise to pack this program. It would be cleaner to pass the initial conditions as variables allowing this program to be a stand-alone module. The initial conditions are: change, time, resistance, capacitance and voltage.

AUTHOR: George Dorner

NAME: ResRatIHC

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$176

DESCRIPTION: This routine finds resistors for a specified ratio and specified percentage tolerance limits.

USE: resrat1hc <cr> self prompting

AUTHOR: Carl R. Kreider

NAME: network

TYPE: Electronics

LANGUAGE: Basic09

SIZE: \$114E

DESCRIPTION: Electronic circuit network design/analysis tool. There was one error in the program that is easily fixed. Change:

```
IF HCS="Y" OR HCS="y" THEN
    OPEN #outpath,"/P1"
```

to:

```
IF HCS="Y" OR HCS="y" THEN
    OPEN      #outpath,"/
P1":WRITE
```

NOTE: /P1 could be /P depending on what your printer driver is set to
USE: network <cr> the program prompts you for a filename if you have a previously saved ".net" file or for manual data entry. You have the option of editing data points but not of saving them from this program. I like the feature of using a data file. One cardinal rule about computers and data, you should never have to enter the same data into a computer by hand more than once. This feature can change any program into a very powerful routine. I did notice that this program will not evaluate DC circuits so don't try using a frequency of zero hertz! The output (either CRT or hardcopy) gives you output frequency in hertz, gain in decibels, and the associated phase angle in degrees. You have a choice of either a linear scale or a log scale for frequencies. This is a great tool for designing filters and such. It handles passive as well as active devices such as resistors, capacitors, inductors, field effect transistors, NPN transistors, and operational amplifiers. The help file "network-how" is very comprehensive. Several example ".net" files are included to provide you with extra help in understanding how the program works.

AUTHOR: Carl R. Kreider

Based on article in EDN Magazine; schneider 10/5/27, Niemeyer 2.4.81, and Steincross 9/1/82.

NAME: fast_fourier

TYPE: Math, data analysis/package

LANGUAGE: Basic09

SIZE: \$16FE

DESCRIPTION: This algorithm performs fast fourier transforms and inverse transforms on data (number of samples must be a power of two). Many periodic functions such as square waves or triangular waves are really just a combination of sinusoidal terms. The square wave for example is derived by adding all of the odd harmonics of a sine wave together. The techniques used today to represent such functions was first presented in 1822 by a famous French mathematician and physicist Jean-Baptiste Joseph Fourier. Today these mathematica series are known as Fourier series. The transform is a technique for going from data mapped in the frequency domain to the time domain. You can also go back from the time domain to the frequency domain.

USE: fast_fourier <cr> menu driven you have the option of loading in a file or entering data in by hand. Another neat feature of this program allows you to save your data to a file!

AUTHOR: Greg Morse

NAME: linefit

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$1060

DESCRIPTION: Provides you with three equations that each minimize the X distance, the Y distance, and the perpendicular distance from each data point to a line.

USE: linefit <cr> all data must be entered by hand

AUTHOR: Greg Morse

NAME: normal

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$241

DESCRIPTION: Demonstrates how to generate random numbers which have a gaussian distribution with a chosen mean and standard deviation.

USE: normal <cr> uses two programs rnd_smpl is an important routine that generates sample data for a normal population. Rnd_smpl was derived from "Simulating Sampling from Normal Populations" by Catherine Lilly and Two-Year College Mathematics Readings; Mathematical Association of America, 1981

AUTHOR: George Dorner

NAME: univariate

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$6CF

DESCRIPTION: Statistics Package provides standard deviation, mean, median, minimum and maximum values, range, average deviation, variance, coefficient variance, standard error, skewness, and kurtosis. It also provides population parameters.

USE: univariate <cr> prompts you for input — manual data entry

AUTHOR: George Dorner

NAME: stdev

TYPE: Math, data analysis/statistics

LANGUAGE: Basic09

SIZE: \$2D3

DESCRIPTION: Subroutine, requires a main program to call it. Calculates statistical values of interest for two arrays of up to 2000 points. Provides standard deviation.

USE:

argvar (xsiz,ysiz,x(i),y(i),statres)

xsiz and ysiz are sizes of array

Continued on Page 20

SPECIFICATION FOR SERIAL INPUT/OUTPUT DRIVERS FOR THE TANDY COLOR COMPUTER AND THE OS-9 OPERATING SYSTEM

Downloaded from CompuServe

Section 1.0: Introduction

All of the Tandy supplied serial drivers for the Color Computer are optimized for driving remote and local terminals to the Color Computer "Console". These drivers are also suitable for general purpose serial input-output drivers, but are not optimized in that application. If fact SCFMan itself assumes a terminal and human operator on the acia port(s).

This document is intended to serve as a discussion instrument and tool in deriving a new class of driver for the machine. A prototype driver exists and is found in the DL 7, (database), of the Compuserve OS-9 special interest group, (SIG). This driver was written to implement the Terminal Communications Package, BiTerm, and is identified as "S10". As this document is being created "on-line", any member of the OS-9 community should feel free to input comments, suggestions, and suggestions to the author: Wm. L. Brady, 70126, 267, 4776-B Carmody Court, Harwood Md. 20776, (301) 952-1761, who assumes the maintenance responsibility.

Section 2.0: Applicable documents

OS-9 Tecnical Information Manual and addendums Microware Systems/Motorola/Tandy Corporation(s).

Section 3.0:

Preliminary Requirements

1. The driver must conform to all applicable OS-9 conventions and standards extant for device drivers, in regards to position independent code, uses of the interrupt facility and device descriptor format, including non-fixed device addresses. However, the driver shall not, necessarily, mimic the operation of all bytes in the device descriptor format, and may re-define any byte that is not required for correct execution of system modules above the driver level, (SCFMan).
2. The driver shall use interrupts to signal the system and user software of each ASCII character received, except for, (when available), DMA, (direct memory access), devices in which case the interrupt shall be generated when the port hardware requires service by the user software. For DMA devices the interrupt shall be user selectable as to generation after timeout, (user specified), or buffer full.

3. The receive buffer shall be user selectable, and shall not be less than 64 characters nor more than 2048 characters.

4. The Transmitter function of the driver may or may not be interrupt driven, however, if interrupt driven shall use a separate interrupt handler, loaded into the OS9 IRQ interrupt facility at a lower priority than the receiver interrupt, and meeting the same buffer requirements as 3 above.

5. The driver shall generate user definable, (via the device descriptor, x-off and x-on), flow control characters TOWARDS the host, and shall not respond to the same characters when received FROM the host.

a) At the users option, via a setstat system call, the receiver shall generate an x-off character whenever a user specified character, (usually \$OD, carriage return), is received by the host.

b) At the users option, via a setstat system call, the receiver shall generate an x-off character at a user specified buffer count, as specified in the device descriptor.

c) The driver shall automatically de-select method a) if method a) is active and method b) is selected, and vice-versa. The driver shall guarantee that only one mode is active at any given time. The selection of one method without deselecting the other shall not cause an error to be generated by the driver.

d) If the driver has sent an x-off, it shall not resend it, and shall send x-on if either a user settable, (via the device descriptor), buffer count is reached, or if a getstat \$01 is executed or a read is attempted against an empty buffer.

5. The driver shall completely configure the port hardware on the INIT system call. All information used by the driver shall be present in the device descriptor at the time of this call.

6. The driver shall completely re-configure the port, after the INIT system call by the SS.ComST setstat call \$28. All information used by the driver shall be present in the device descriptor at the time of this call.

7. The driver shall provide, as a minimum, three "filter" capabilities. A filter is defined as a criteria which a received byte must meet prior to its being placed in the input buffer by the driver.

a) One filter which shall delete all characters GREATER THAN, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

b) One filter which shall delete all characters LESS THAN, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

c) One filter which shall delete all characters EQUAL TO, (ASCII numeric value), a byte specified in the device descriptor. (00 signifying filter off).

8. The driver shall monitor detected parity errors and overruns on the port hardware.

a) Overruns shall be processed by the interrupt handler by inserting a user specified, (via device descriptor), graphics character in the input buffer. The driver shall not generate an error for this condition.

b) Parity errors shall be processed as any other character, however, the driver may be

designed to replace any errored character with the same graphics character as in a) above.

9. The driver shall reset its buffer pointers at user request via a setstat call. This is to allow the user to "replay" the entire buffer contents for a repeat screen display function or a "quick print feature".

Section 3.A: Optional Driver Features

1. The driver may have the capability to open and send characters directly to the standard output path, (screen), however, this capability shall be controlled by setstat calls, and shall be deselectable. This capability shall not replace any of the capabilities in section 3.0.
2. The driver may have the capability to send characters directly to one or two user specified path(s), (method of specification TBD), however, this path shall not use the primary driver buffer or buffer pointers. This feature shall be controllable by setstat calls, and, if implemented, meet the filter requirements in 3.7 a, b and c above (this feature would support on-line printing and downloads). It is recommended that feature automatically select an x-off method as specified in 3.4 above.

Table 3..0:

Additional Replacements to the "T" class of Device Descriptors

Name	Relative Address	Size (bytes)	Use
IT.FLB	tbd	1	Greater than filter byte
IT.FLL	tbd	1	Less than filter byte
IT.FLE	tbd	1	Equal to filter byte
IT.BAD	tbd	1	Bad Character (overrun) Graphics character
IT.BUF	tbd	1	Buffer size selection (table entry)
IT.EOR	\$1D	1	Same as T class driver but different usage see 3.5.a
IT.MIN	tbd	2	Buffer near empty count for xon
IT.MAX	tbd	2	Buffer near full count for xoff
IT.BF1	tbf	2	Buffer 1 (DMA devices only)
IT.BF2	tbd	2	Buffer 2 (DMA devices only)

Don't Miss
RAINBOWfest Princeton
October 17-19, 1986!



POSITIVE ACTION VS. NEGATIVE COMMENTS

by Dale Puckett, Director-at-Large

We OS-9 users live in a small universe. There aren't too many of us. Sure, our favorite operating system appears to be coming out of the underground and capturing its own spot in the popular press this

year. But, if we all really want to see it succeed and become the MD-DOS equivalent in the 68000 marketplace, we must all do our part.

"What can I do," you ask? Become an OS-9 evangelist.

That's a natural. I'll bet you are already doing it. If I were a fly on the wall, I'd hear you telling all your MS-DOS friends about the many great OS-9 features you have grown to know and love.

OK, we're all marketing OS-9 to the best of our abilities, but, are we covering all bases? I don't think so because we don't have everything we need. If OS-9 is to succeed in the consumer marketplace it must have a reason for being. There must be hundreds of application programs that do jobs that people are tired of doing by hand.

To make this happen we must encourage programmers within our ranks. We must salute them when they move the state of the art forward. We must encourage them to make bold steps forward with innovative techniques rather than discourage them with our criticism.

I say these things after reading the mail on CompuServe's OS-9 SIG and on Delphi's Color Computer SIG. There are a lot of helpful people on both of these electronic bulletin boards, but there is also much criticism. This criticism, especially when combined with low sales and minimal profit, discourages programmers from writing new programs. As a result, we all suffer.

People criticize Tandy for making a business decision when they designed their implementation. Then, they criticize Microware for delivering the product the customer ordered. They don't stop to realize one important basic of the business world — the customer is always right.

Frankly, Tandy had a good reason for every "feature" they put in

the Color Computer implementation of OS-9. We may or may not agree with the selection made by Tandy's designers but we must realize that no business decisions are made lightly. And to be quite honest, we must also realize that if Microware hadn't been willing to deliver the product that Tandy wanted, Tandy probably would have chose another operating system for the Color Computer. Then where would we be?

In the old days the story read, "Those who can, do. Those who can't, teach." If recent experience is any indication that old parable might be reworded ever slightly. "Those who can, do. Those who can't, criticize."

In case you are wondering, this column was born as the result of several interchanges on one of the bulletin boards above. Another use took me to task for being positive and the debate went back and forth for several weeks.

I wasn't sorry for being positive then. I am not sorry now. The world would be a lot nicer place to live if more people had a positive attitude. Because of this belief, I work very hard to maintain and project a positive attitude even though it is very hard at times.

As I told my adversary, everyone deserves a pat on the back every once in awhile — especially when he is not getting rich in the market he is supporting out of pure love.

If this positive attitude theory seems like a lot of hogwash to you, I challenge you to look around the business world. I think you'll find that the companies that project a positive "can do" attitude are the ones that are getting ahead. Organizations run by managers who try to think of reasons not to do something are falling like flies.

The bottom line, if you know a better way to do something, do it. If it's something everyone can use, sell it. If not, share it. But, do it. And, don't put the other guy down just because his approach is a little different.

OS-9

Application for membership in the OS-9 Users Group

Last: _____

First: _____

Middle: _____

Street: _____

City: _____

State: _____ ZIP: _____

Phone: (_____) _____

Delphi ID: _____

Computer Make and Model: _____

OS-9 Level: _____ (I) _____ (II) _____ (68K) _____ (CoCo)

Disk Size: _____ (5¼") _____ (8")

Disk Format: _____ (CoCo) _____ (Standard)

Single Sided: _____ Double Sided: _____

of tracks: _____ (35) _____ (40) _____ (80)

Other restrictions, formats, comments, etc: _____

Enclose \$25 to cover the first year's membership in the form of a personal check or money order. Please allow approximately 3-5 weeks for processing your membership. The Users Group has a small staff, and applications are averaging 30 to 50 a week.

Shortly after acceptance of your application for membership, you will receive the current Group newsletter ("MOTD"), and soon after, the "starter" diskette, UG Disk # 0, with software of the type useful in getting you started with both OS-9 and the Users group, including a modem program to assist you with CompuServe access while under OS-9. Additional diskettes may be purchased at a cost of \$5 each to cover the cost of media and postage.

Mail your application and other correspondence to:

The OS-9 Users Group
ATTN: (department)
9743 University Avenue
Suite 330
Des Moines, IA 50322

Where (department) is President, MOTD Editor, Librarian, Membership, etc.

Membership dues can be charged to your VISA or MasterCard. Simply enter your card information below and sign this form.

Card type: VISA _____ MasterCard _____

Account #: _____

Expires: _____

Is this a Renewal? Yes _____ No _____

LETTER TO THE EDITOR

Gentlemen;

We have been discussing on the OS-9 SIG a new class of serial I/O driver for OS-9. I have put up a "straw-man" spec in d17. (SSPEC.DOC). You may want to pick it up for publication in MOTD.

Bill Brady

You'll notice SSPEC.DOC in this issue of the MOTD. It's always fun for me to be able to respond to people that have something positive to add to the MOTD. Constructive enhancements to the world of OS-9 are what will help it to grow into THE operating system of the future.



at RainbowFest Princeton
October 17-19, 1986

Meet a Hero (of OS-9)

Get answers

Come see: Dale L. Puckett, Director-at-Large
Brian A. Lantz, President
Bill Turner, Vice President
Bruce Warner, Editor, MOTD

Make friends

Have breakfast Sunday morning with Frank Hogg of
Frank Hogg Laboratory

. . . get the straight answer for what's
coming down the line.

Breakfast Tickets - \$14.00

Get advance tickets by writing:

Share knowledge

Renew membership

OS-9 Users Group
ATTN: Breakfast
Suite #330
9743 University Ave.
Des Moines, Iowa 50322

Sign up a friend

Solve problems

PIPE OF THE MONTH

by Kevin Kuehl

In this article I am going to show you how to build a simple but powerful database manager using only six common utilities. The procedure is very simple and uses only cat, del, grep, rename, sort, and tr in three steps with one being a pipeline.

In my example I will be building a list of addresses in a file called 'addrbook.' The file has this format:

```
lastname, firstname~address~city, state zip~<cr>
```

Step one is to type this statement:

```
OS9: cat > tempfile<cr>
```

```
Kuehl, Kevin~806 Division Road~Valparaiso, IN
```

```
46383~<cr>
<eof>
```

With this I have used cat to put my address in a file named 'tempfile.' But why are the tildes in the line? The pipeline needs some character to serve as a separator which will become a carriage return when run through. Why have I used the tilde? Because I have yet to see it in an address. Actually the separator can be any character you want, but I have found the tilde to work best. Why have I included a tilde at the end of the line when tr will print the last carriage return? If you have a number of lines that the search string matches, the extra tildes add

one blank line in between them. So really the only reason it is there is for aesthetics.

The next step is to append this information to the end of the file 'addrbook.' If this file doesn't already exist, then you can simply change the previous command to send its output to that file. Otherwise you should use cat, del and rename like so:

```
OS9: cat addrbook tempfile >
newfile; del addrbook
tempfile; rename newfile ad-
rbook<cr>
```

This updates 'addrbook' to include the new information included in 'tempfile.' In actual use, I would keep this line in a shell script with a simple name such as update. If you have one, a program such as cp which appends to the end of a file if it already exists is a time saver. I have one and I just type:

```
OS9: cp tempfile addrbook<cr>
```

and the computer does the work. Also 'tempfile' can include any number of new additions — not just one as in my example.

The final step is to get a printable output given a search string. My address can be found by typing:

```
OS9: grep Kevin addrbook ! grep
Kuehl ! tr~/13<cr>
```

Why did I use grep twice? With

only one I must search for either 'Kuehl' or 'Kevin,' but not both. This way I am able to make sure I get my name only. If I were to search for 'Kuehl, Kevin,' grep would become confused and look for 'Kuehl,' in the file 'Kevin' — definitely not what we want. The first grep matches all 'Kevins' and the second matches only 'Kevin Kuehls' then sends its output to tr which changes all the tildes to carriage returns and gives us:

```
Kuehl, Kevin<cr>
806 Division Road<cr>
Valparaiso, IN 46383<cr>
<cr>
```

Why did I put the name at the beginning of the line last name — first, first name — last? So that a sort program would work on the name correctly. If I were to type:

```
OS9: sort addrbook ! grep IN !
tr~/13<cr>
```

I would be presented with a neatly sorted listing of all the people in 'addrbook' who live in Indiana.

I hope this short article has helped you put to use some common utilities and maybe even given you some ideas for others. The procedure I have laid out should work with any set of utilities since I have not used any options except for the decimal representation of a carriage return in tr.

OS-9

PATCHING STRIP

by Jay Truesdale

Here is a copy of a message I posted in reply to Dave K. about the strip utility:

Dave, in regards to my previous message about the strip utility. The problem with the "meaningless characters" appears to be fixed by changing the LDX #CHAR to LEAX CHAR,U as I suggested.

I believe the "meaningless characters" are actually control characters, which in my case, cause my WordPak II to perform functions like cursor addressing, etc. The description of the I\$Write system call in my (Color Computer) OS-9 manual says that register X is to contain the starting address of the data string to

be written. From looking at the object code generated by the assembler, it looks to me like the instructions LDX #CHAR would load register X with the value \$000D. The following I\$Write system call would then proceed to write the two bytes that are at location \$000D, which could be just about anything, and in this case are probably seen as control characters by my WordPak II.

After reading this, I realized it would make more sense if you read Dave's letter; it is on the OS-9 SIG on Subtopic 3.

I hope this is correct and makes sense, if not please let me know.

OS-9

PLACING AN AD IN THE MOTD

There are still a lot of people out there that want to advertise in the MOTD. So what do they do?

Well, here's the deal! You start out by getting your ad made up (called camera ready copy). When that is completed, you submit your ad with a check made out to the OS-9 Users Group and send both to:

Editor, the MOTD
c/o Bruce N. Warner
14503 Fullerton Road
Dale City, VA 22193-2034

The price for advertising in the MOTD is based on the size, location and number of colors (colors limited to black and red). This chart explains more completely.

SIZE	REGULAR		SPECIAL (back cover)	
	1-Color	2-Colors	1-Color	2-Colors
Full Page	\$400	\$480	\$500	\$600
Half Page	\$200	\$240	\$250	\$300
Quarter Page	\$100	\$120	\$125	\$150
Eighth Page	\$ 50	\$ 60	\$ 62.50	\$ 75

Rates are higher for the special issue that will be inserted in RAINBOW magazine. These rates will be published at a later date.

You'll have to make sure that you've included your camera ready copy and a check for payment made out to the OS-9 Users Group. The deadline for entries is the first of every month for the issue beginning the following month.

A PATCH FOR MORE STORAGE

by Steve Bjork and Pete Lyall

(Editor's Note: The following information is provided courtesy of CompuServe.)

Getting more out of (or perhaps on to) the DeskMate disk can be as easy as making a few patches to the CCDISK module. Here are the necessary offsets and values.

Offset	Old	New	
1f9	13	10	6 ms step rate
1ff	22	02	Shorten commands delay
209	01	01	Drive 0 (front of DS 0)
20a	02	41	Drive 1 (back of DS 0)
20b	04	02	Drive 2 (front of DS 1)
20c	40	42	Drive 3 (back of DS 1)
2e8	03	00	6 ms restore
341	60	84	New CRC value
342	da	df	New CRC value
343	8c	79	New CRC value

MY CADILLAC FOR THE COCO

by Richard Johnson, CompuServe OS-9 SIG

This driver was written out of a need to support the extra function keys on my keyboard. I have the "Deluxe CoCo" keyboard which has a true control key, an alt key, and an F1 and F2 key. It has the arrow keys arranged in a cluster and has the clear key positioned by the '@' key. I realized that PBJ wasn't going to support non-standard keyboards, so I wrote this driver. It will support any keyboard that has extra function keys mapped into the four extra slots in the standard keyboard's matrix. It also allows all keys to be reprogrammed to return new values.

Any value may be returned except \$FF and \$FD, which is used by the driver internally, \$FE, which is used for the Caps Toggle value instead of the \$1F that both the standard CCIO and WordPak drivers use. I changed it so that the keyboard could send all ASCII values (\$00 through \$7F). This allows the keyboard to emulate any keyboard that

sends single character codes for key depressions. Also, those who want a DVORJAK keyboard layout can have it just by reprogramming the key tables. As released the keyboard works exactly as before with the following exceptions:

the CLEAR key sends a \$08 (or Backspace) in both the normal and shifted modes. Ctrl-Clear sends a rub-out (\$7F).

the CTRL works as the control key the ALT is now an escape key and sends \$1B in all modes

the F1 key sends a (unshifted) a '\', (shifted) a '{', and (ctrl) a '}'.

the F2 key sends a (unshifted) a '-', (shifted) a '[', and (ctrl) a ']'.

I chose these for convenience in 'C' programming. Using the reprogramming option these keys as well as 1 through 0 and :-@ plus the <BREAK> and <Spacebar> keys can be changed. Below is a chart that shows the positions of all the keys in the Keyboard matrix.

KEYBOARD TABLE
COCO deluxe keyboard matrix

#		A	B	C	D	E	F	G
8	H	I	J	K	L	M	N	O
16	P	Q	R	S	T	U	V	W
32	X	Y	Z	ua	da	la	ra	sp
48	#	1	2	3	4	5	6	7
56	\$	9	:	;	,	-	.	/
64	en	cl	br	ctr	alt	f1	f2	sh

** ----- used for the joysticks -----
1 2 3 4 5 6 7

LEGEND :

ua = uparrow	en = enter
da = downarrow	cl = clear
la = left arrow	br = break
ra = right arrow	ctr = control
sp = space bar	f1 = function key 1
sh = shift	f2 = function key 2

How To Reprogram Keys

The keys are arranged as in the Matrix in the keyboard table with the '@' key followed by the <ua> (uparrow) through <f2> keys. There are three tables, each containing a slot for each key. To find a key in any table use the locations in the above table. The keys may be reprogrammed by using OS-9's display command or by SetStat calls. A key's current value may be obtained by a GetStat call. The Control and shift keys can only be changed in the driver source code and changing their values in the tables will have no affect on them.

There are three Display codes, each corresponding to one of three tables. The first is \$1D for the

Unshifted keys, the second is \$1E for the Shifted keys and finally \$1F for the Control keys. The use of the Display command:

Display 1F {value} {key offset}

Where "\$1F" could be either \$1F, \$1E or \$1D, for the respective table; "value" is the new value of the key and "key offset" is the offset of the key into its table. All values must be in Hexidecimal and be of one byte in length.

Using the Status Calls works in similar ways except the codes for each table are 150 unshifted, 151 for shifted and 152 for control. The use of both the SetStat and GetStat calls is described below.

For Setstat -

Entry

A = Code (15# dec)
B = New Key Value
X = offset into the Table

Returns

B = #
cc is clear

A GetStat for the Function keys is also included -

Entry

A = Code (15# dec)
X = Offset into table

Returns

on success -

A = key value
cc is clear

on error -

B = error code
cc is set

Compatibility With Other Drivers

This Driver, WordPakC, only works in a Radio Shack compatible mode. Some O-Pak compatibility has been implemented. It is not compatible with the standard WordPak mode. All the functions of the Radio Shack CCIO that relate to the text screen have been implemented and the WordPak driver 3.1 RS functions have also been added. The last page is a chart showing display codes and their effects. The cursor may be changed either by using sequences or by the RS compatible "05 nn" codes. Note however that since the WordPak is only a monochrome display board the shape and blink rate are changed instead of the color.

Status Calls (Getstat) -

All of the CCIO codes are supported (including joysticks) except the following two — SS.DSTAT, which returns graphic screen data, and SS.ALFAS, which is dependent

on a memory mapped display. The following O-Pak calls are supported as well —

CODE 131 — last displayable character — always returns \$7F in A.

CODE 132 — O-Pak screen size call. Don't use this call, included for software compatibility. A = columns, B = rows on screen (80,24).

CODE 134 — Cursor Status. A = cursor blink rate, B = cursor type (Block or UL)

CODE 135 — Overstrike Mode — always in overstrike mode. A = 0.

As the standard CCIO is never likely to support these calls, I have only included them for completeness. I recommend that you never use them in your own programs to maintain compatibility with CCIO.

How to Install WordPakC and Term

First assemble the files WordPakC.asm and TermC.asm into executable files. Use the following line:

OS9: asm WordPakC.asm o=/d0/WordPakC #30k;asm TermC.asm o=/d0/TermC #20k

If you have any WordPak drivers, just use their install procedure, except substitute WordPakC for WordPak in the file Bootfile. If you have Sdisk, do the same for CCIO and use the TermC descriptor Term for WordPakC. (TermC will appear as Term when loaded so no patching of Init is needed and new BootFiles may be created using it; TermC is exactly like the PBJ Term.)

If you would like a Write only version to use along with CCIO, you can carefully delete all keyboard functions, the keyboard table, the Getstat and Setstat Codes for key-

board use and the diplay code for the Function Key changing and make the driver a write only driver. Then use the following for Read and Setstat:

```
READ
PUTSTAT
clrb
rts
```

And the driver will only be capable of display. This will eliminate the dread keyboard lock up that comes from trying to use both the PBJ drivers and CCIO at the same time. Change the name of TermC from "Term" to "WP" so that it doesn't conflict with the CCIO's Term.

I have done the best job I could on this driver, however, I'm not the best assembly language pro-

Continued on Page 12

NATIONAL COMPUTER CONFERENCE

by Bruce Warner

Right: Bruce Campbell, President of mpd USA, Inc., demonstrates the latest version of Sculptor (The Fourth Generation Programming System) in the Gimix Booth.



Right: OS-9 running on an IBM PC could make the PC a real computer. The TLM 68000 OS-9 board in an IBM XT shows that the IBM does have more uses than keeping the door open.



Above: The man in the jacket is laughing because everyone thinks you have to have an IBM PC to get anything done. In fact, he's using an Apple IIe with the MicroTRENDS implementation of Jonathan. This could be the start of a real computer. Top Right: Virtually unlimited power is possible inside a small case. At the Microware booth, the GMX-020 single-board computer was displayed to point out the possibilities of having real computer power in a limited office space environment. Bottom Right: If it wasn't for TLM's 68000 OS-9 board, you might have never seen Bill Moore at an IBM PC keyboard, but innovation can change almost anything.



LAS VEGAS — If there's one thing you can say about the National Computer Conference here, it's that OS-9 was one of the VERY few new items on the agenda. It looks as if the rest of the industry was out to lunch while OS-9 was taking the leading edge of what's new.

For those of us that read the MOTD, there's nothing new about CD-I ROM, but for the many attendees at the conference, CE-I ROM was absolutely new and innovative. My greatest joy was laughing at the MS-DOS machines that failed to meet the abilities of OS-9, even 6809 Level 1 OS-9.

Seeing OS-9 in the Microware booth was no surprise, but when I looked in some of the other booths and found OS-9 alive and well, I was elated.

Gimix, an Old Hand at OS-9

The Gimix booth was displaying OS-9 on their 68000 machines, including a new 68000 single board computer that fits in the case of an IBM-PC. While it's not an identical footprint of the IBM motherboard, it is a well-designed computer that allows you all the capabilities of OS-9 without the need for an expensive, custom-designed power supply and case. The Taiwan specials were custom-made for this little jewel.

Show Special Carried to UG From Data-Comp

One of the specials of the show has been extended to the Users (group through October 1). Thanks to Don Williams of Data-Comp, you can order the Mustang-020 — 100 Golden Eagle Special through October 1, 1986 at just \$3,998.00. This includes the following features:

- GMX Micro-20MC68020 SBC (for the MC 68020 16.67 Mhz CPU add \$300)
- DATA-COMP Metal Enclosure with switching power supply
- DS, DD, 80 Track Floppy Disk Drive
- 25 MByte Winchester Hard Disk Drive
- XEBEC 1410A Hard Disk Controller

- All Special Cables-Factory Assembled-48 Hour Burn-in
- Choice of Operating System: (OS-9, UniFLEX) (Both the OS-9 and UniFLEX operating systems are UNIX System V, C level compatible.)
- 4 Serial Ports, Pre-Wired

Don also informs me that you must make payment in full with your order to qualify for this discount. Add to this an additional 10 to 70 percent discount on all 68XXX software to registered owners. You can reach Data-Comp at 5900 Cassandra Smith Road, Hixson, TN 37343 or call at (615) 842-4600.

Sculptor Keeps Getting Better

Another novelty at the show was the update of Sculptor. It's now faster than ever, and still lives up to all its previous flexibility. The greatest increase is in its ability to execute statements without having to reload a shell for each execution. This alone greatly enhanced the speed of Sculptor.

CD-I Conference

The CD-I ROM conference was another session worth attending. This one really put us in beyond the big boys. The funniest question came from an attendee that asked what would happen if Big Blue were to try to get involved. Logic dictates the answer to this one. Big Blue is not a developer of operating systems. Neither are they the hardware developers of compact disk players. The story of CD-I ROM lies between these two marketeers, and Big Blue is not in the drivers seat. Facts are facts, and CD-I ROM is based on OS-9.

The balance of the show was fairly bland, leaving most of us a little cold. I was somewhat disappointed with the overall show. Luckily, I wasn't the only one to have these feelings. I was more than a little happy to repack my bags and head for the airport.



An OS-9 hopeful (L.) talks with Jeanne Kaplan and Bill Moore about the advantages of OS-9. (The visitor had her back pointing directly at the IBM booth.)



If the case looks a lot like what you see from Big Blue, don't fret. Inside this case lies the heart of a real single-board computer.



Above: There's almost no limit to the number of 68020 OS-9 computers on the market, as DATA-COMP showed off their implementation of the GMX-020 in the Gimix booth. Left: Microware founder Ken Kaplan speaks to the many visitors at the CD-I ROM press conference. For those of us who have followed the story of CD-I ROM in the MOTD there wasn't much new to hear about, but to the world as a whole, CD-I was news for sure.



Left: The complete Collier Encyclopedia is accessible from a single compact disk. This is the forerunner of the industry shuttering CD-I ROM. The version presented in the Philips booth at NCC was running on the Apple IIe.



Above: Is there any limit to the possibilities of OS-9? Evidently not, as OS-9 was running on the Motorola System 1000. Right: While the Atari ST was running Hack, Bill Moore decided to run another task on the remote terminal to demonstrate its flexibility to a show visitor. Even Atari gets a chance to grow out of the game world when you offer them OS-9 as an operating system.



Right: Bruce Campbell, President of mpd USA, Inc., demonstrates the latest version of *Sculptor* (The Fourth Generation Programming System) in the Gimix Booth.



Right: OS-9 running on an IBM PC could make the PC a real computer. The TLM 68000 OS-9 board in an IBM XT shows that the IBM does have more uses than keeping the door open.



Above: The man in the jacket is laughing because everyone thinks you have to have an IBM PC to get anything done. In fact, he's using an Apple IIe with the MicroTRENDS implementation of *Jonathan*. This could be the start of a real computer. Top Right: Virtually unlimited power is possible inside a small case. At the Microware booth, the GMX-020 single-board computer was displayed to point out the possibilities of having real computer power in a limited office space environment. Bottom Right: If it wasn't for TLM's 68000 OS-9 board, you might have never seen Bill Moore at an IBM PC keyboard, but innovation can change almost anything.



Show Special Carried to UG From Data-Comp

One of the specials of the show has been extended to the Users Group through October 1. Thanks to Don Williams of Data-Comp, you can order the Mustang-020 — 100 Golden Eagle Special through October 1, 1986 at just \$3,998.80. This includes the following features:

- GMX Micro-20MC68020 SBC (for the MC 68020 16.67 Mhz CPU add \$300)
- DATA-COMP Metal Enclosure with switching power supply
- DS, DD, 80 Track Floppy Disk Drive
- 25 MByte Winchester Hard Disk Drive
- XEBEC 1410A Hard Disk Controller



An OS-9 hopeful (L.) talks with Jeanne Kaplan and Bill Moore about the advantages of OS-9. (The visitor had her back pointing directly at the IBM booth.)

- All Special Cables-Factory Assembled-48 Hour Burn-in
- Choice of Operating System: (OS-9, UniFLEX) (Both the OS-9 and UniFLEX operating systems are UNIX System V, C level compatible.)
- 4 Serial Ports, Pre-Wired

Don also informs me that you must make payment in full with your order to qualify for this discount. Add to this an additional 10 to 70 percent discount on all 68XXX software to registered owners. You can reach Data-Comp at 5900 Cassandra Smith Road, Hixson, TN 37343 or call at (615) 842-4600.

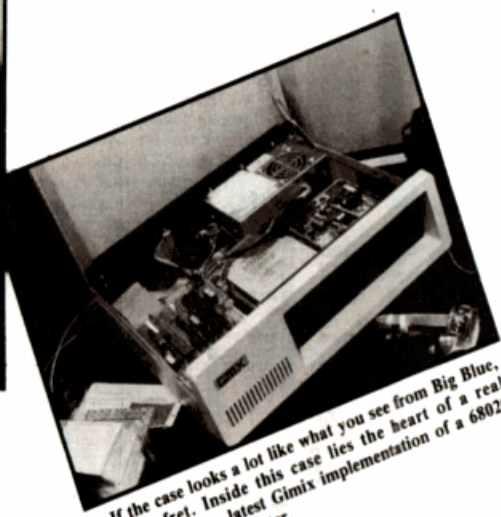
Sculptor Keeps Getting Better

Another novelty at the show was the update of *Sculptor*. It's now faster than ever, and still lives up to all its previous flexibility. The greatest increase is in its ability to execute statements without having to reload a shell for each execution. This alone greatly enhanced the speed of *Sculptor*.

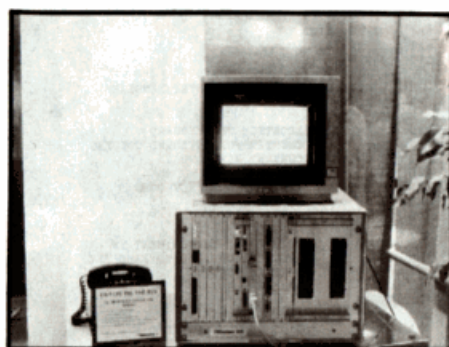
CD-I Conference

The CD-I ROM conference was another session worth attending. This one really put us in beyond the big boys. The funniest question came from an attendee that asked what would happen if Big Blue were to try to get involved. Logic dictates the answer to this one. Big Blue is not a developer of operating systems. Neither are they the hardware developers of compact disk players. The story of CD-I ROM lies between these two marketers, and Big Blue is not in the drivers seat. Facts are facts, and CD-I ROM is based on OS-9.

The balance of the show was fairly bland, leaving most of us a little cold. I was somewhat disappointed with the overall show. Luckily, I wasn't the only one to have these feelings. I was more than a little happy to repack my bags and head for the airport.



If the case looks a lot like what you see from Big Blue, don't fret. Inside this case lies the heart of a real computer, the latest Gimix implementation of a 68020 single-board computer.



Above: Is there any limit to the possibilities of OS-9? Evidently not, as OS-9 was running on the Motorola System 1000. Right: While the Atari ST was running *Hack*, Bill Moore decided to run another task on the remote terminal to demonstrate its flexibility to a show visitor. Even Atari gets a chance to grow out of the game world when you offer them OS-9 as an operating system.



NATIONAL COMPUTER CONFERENCE

by Bruce Warner

LAS VEGAS — If there's one thing you can say about the National Computer Conference here, it's that OS-9 was one of the VERY few new items on the agenda. It looks as if the rest of the industry was out to lunch while OS-9 was taking the leading edge of what's new.

For those of us that read the MOTD, there's nothing new about CD-I ROM, but for the many attendees at the conference, CE-I ROM was absolutely new and innovative. My greatest joy was laughing at the MS-DOS machines that failed to meet the abilities of OS-9, even 6809 Level 1 OS-9.

Seeing OS-9 in the Microware booth was no surprise, but when I looked in some of the other booths and found OS-9 alive and well, I was elated.

Gimix, an Old Hand at OS-9

The Gimix booth was displaying OS-9 on their 68000 machines, including a new 68000 single board computer that fits in the case of an IBM-PC. While it's not an identical footprint of the IBM motherboard, it is a well-designed computer that allows you all the capabilities of OS-9 without the need for an expensive, custom-designed power supply and case. The Taiwan specials were custom-made for this little jewel.

From Data-Comp

One of the specials of the show has been extended to the Users Group through October 1. Thanks to Don Williams of Data-Comp, you can order the Mustang-020 — 100 Golden Eagle Special through October 1, 1986 at just \$3,998.80. This includes the following features:

- GMX Micro-20MC68020 SBC (for the MC 68020 16.67 Mhz CPU add \$300)
- DATA-COMP Metal Enclosure with switching power supply
- DS, DD, 80 Track Floppy Disk Drive
- 25 MByte Winchester Hard Disk Drive
- XEBEC 1410A Hard Disk Controller



Above: There's almost no limit to the number of 68020 OS-9 computers on the market, as DATA-COMP showed off their implementation of the GMX-020 in the Gimix booth. Left: Microware founder Ken Kaplan speaks to the many visitors at the CD-I ROM press conference. For those of us who have followed the story of CD-I ROM in the MOTD there wasn't much new to hear about, but to the world as a whole, CD-I was news for sure.



Left: The complete Collier Encyclopedia is accessible from a single compact disk. This is the forerunner of the industry shattering CD-I ROM. The version presented in the Philips booth at NCC was running on the Apple IIe.

COCO CADILLAC

Continued from Page 9

grammer and I'm sure that this Driver could stand improvement. So if you can improve it go ahead and do it! Then kindly drop me a note on the OS-9 SIG on Compu-Serve. My number is 74116,3201 so I can see what you've done. Also be sure to leave a copy in the library.

.PO #

SCREEN DISPLAY CODES

Hex	Decimal	Screen Function
#1	#1	Home Cursor
#2 x,y	#2 x,y	position Cursor(x = Column + \$2#, y = rows + \$2#)
#3	#3	Erase Line
#4	#4	Erase to End of Line
#5 n	#5 n	Change Cursor (n = 12 to 42) (Operates differently, See Text)
#6	#6	Cursor Right
#7	#7	Ring Bell
#8	#8	Cursor Left
#9	#9	Cursor Up
#A	#A	Cursor Down
#B	#B	Erase to End of Screen
#C	#C	Home Cursor and Clear Screen
#D	#D	Carriage return
1B 41	27 65	Erase to End of Line
1B 42	27 66	Erase to End of Screen
1B 45	27 69	Insert a Line
1B 46	27 70	Delete a Line
1B 47	27 71	Shift Right
1B 48	27 72	Shift Left
1B 53 2#	27 83 32	Set Normal Video
1B 53 21	27 83 33	Set Inverse Video
1B 56	27 86	Block Cursor
1B 57 r	27 87 r	Cursor Blink Rate (2# = steady, 21 = Slow, 22 = Fast, 23 = Invisible)
1B 76	27 118	Underline Cursor
1D v o	29 v o	Change key at offset 'o' in the unshift key table to value in 'v'
1E v o	3# v o	Change key at offset 'o' in the shifted key table to value in 'v'
1D v o	29 v o	Change key at offset 'o' in the control key table to value in 'v'

(*) - Indicates non CCIO / RS-089 1.#1.## compatible codes
(**) - indicates unique to this driver only.

This driver is for Educational purposes and is not to be distributed or sold as part of any commercial package.

#001	NAM	WORDPAK REPLACEMENT DRIVER
#002	TTL	DEFINITIONS
#003	*	
#004	*	WORDPAK DRIVER
#005	*	
#006	IFP1	ENDC
#007	ENDC	
#008		
#009	TYPE	DRIVER+OBJECT
#010	REVS	REENT+2
#011	INPSIZ	SET \$#
#012	MOD	WORDP1,WPAM,TYPE,REVS,ENTRY,MEMSIZE
#013	ORG	V.SCF
#014	* VARIABLES FOR KEYBOARD ROUTINES	
#015	KPORT	RMB 2 LOCATION OF KEYBOARD
#016	COLCNT	RMB 1 COUNTER FOR KEYBOARD COLUMNS
#017	KEY	RMB 1 CURRENT KEY DOWN
#018	LASTKY	RMB 1 LAST KEY DOWN
#019	KEYVAL	RMB 1 VALUE OF LAST KEY DOWN
#020	CAPLOK	RMB 1 CAPLOCK FLAG
#021	SHFLAG	RMB 1 SHIFT FLAG
#022	CTRFLG	RMB 1 CONTROL KEY FLAG
#023	BUFIN	RMB 1 POINTER TO BUFFER:NEXT IN
#024	REPSPD	RMB 1 ANOTHER FOR READ TO FIND KEY
#025	REPWAT	RMB 1 SPEED OF REPEAT
#026	WAIT	RMB 1 WAIT COUNT
#027	* VARIABLES FOR VIDEO ROUTINES	
#028	ESCONT	RMB 1 COUNTER FOR KEYS IN VIDEO ES
#029	ESCTFLG	RMB 1 FLAG THAT HOLD VIDEO ESC CHAR
#030	ESCBUF	RMB 1 HOLDS 2ND CHAR IN ESC SEQ
#031	CURLIN	RMB 1 CURSOR LINE
#032	CURCOL	RMB 1 CURSOR COLUMN POS
#033	INVFLG	RMB 1 INVERSE VIDEO FLAG
#034	DSTART	RMB 2 DISPLAY START
#035	WORK	RMB 2 A PLACE FOR SHORT TERM STORA
#036	INPBUF	RMB INPSIZ INPUT BUFFER (IN CASE OF OVE
#037	MEMSIZE	EQU .
#038	FCB	READ.+WRITE.+EXEC.
#039	TTL	WORDPAK REPLACEMENT DRIVER
#040	FCB	"WordPakC"
#041	FCB	1
#042	ENTRY	
#043	LBR4	INIT
#044	LBR4	READ
#045	LBR4	WRITE
#046	LBR4	GETSTAT
#047	LBR4	PUTSTAT
#048	LBR4	TERM
#049	PHS	DP,CC
#050	ORCC	\$55#
#051	CLRA	SET DP
#052	TFR	A,DP
#053	STU	D.KBDSTA
#054	LTX	D.IRQ
#055	STX	D.ALTIRQ
#056	LEAX	POLKEY,PC
#057	STX	D.IRQ
#058	LTX	\$55#
#059	LTX	\$55#
#060	LTX	\$55#
#061	LTX	\$55#
#062	LTX	\$55#
#063	LTX	\$55#
#064	LTX	\$55#
#065	LTX	\$55#
#066	LTX	\$55#

#067	STX	KPORT,U	PUT IT INTO VARIABLE AREA
#068	CLRB		
#069	STB	CAPLOK,U	NORMAL MODE
#070	STA	SHFLAG,U	SET SHIFT OFF
#071	STA	CTRFLG,U	CLEAR CONTROL FLAG
#072	STD	BUFIN,U	ZERO BUFFER IN AND OUT POINT
#073	STA	\$1,X	SET UP KEYBOARD PIA
#074	STA	\$2,X	
#075	STA	\$3,X	
#076	* SET SCREEN VARIABLES		
#077	STD	DSTART,U	START TO TOP OF SCREEN MEMORY
#078	STD	ESCONT,U	ESCAPE VARIABLES TO ZERO
#079	STD	ESCBUF,U	
#080	STA	CURCOL,U	START OF LINE
#081	STA	INVFLG,U	NORMAL VIDEO
#082	COMB		
#083	STB	\$2,X	SET KEYBOARD PIA
#084	STB	KEY,U	SET KEY BUFFERS
#085	STB	LASTKY,U	
#086	STB	KEYVAL,U	
#087	LDA	\$634	MAKE KEYBOARD REGISTERS
#088	STA	\$1,X	TO R/W
#089	LDA	\$53F	INSTEAD OF DATA DIRECTION
#090	STA	\$3,X	
#091	LDA	\$2,X	
#092	CLRA		
#093	LEAX	WPITBL,PC	INITIALIZE CRTC
#094	LDB	,X+	GET VALUE
#095	STD	[1,U]	STOR INTO CRTC
#096	INCA		
#097	CMPS	\$514	UNTIL ALL DONE
#098	BNS	WPINIT	LOOP
#099	LBSR	HOMCLS	CLEAR/START FRESH
#100	LDD	\$56#	BLOCK CURSOR, SLOW BLINK
#101	STD	CURLST,PCR	
#102	LDA	\$FF23	INITIALIZE PORT FOR BELL
#103	ANDA	\$11111111	SET UP SINGLE BIT SOUND
#104	STA	\$FF23	
#105	LDB	\$FF23	
#106	ORB	\$55#	
#107	STB	\$FF23	
#108	ORA	\$55#	
#109	STA	\$FF23	
#110	CLRB		
#111	CLRA		
#112	PULS	PC,DP,CC	OK-RESTORE AND RETURN
#113	*		
#114	*		
#115	* SCREEN INIT TABLE		
#116	*		
#117	*		
#118	*		
#119	WPITBL	FCB	111,8#,87,6#
#120	WPITBL	FCB	28,5,24,26
#121	WPITBL	FCB	12#,8
#122	WPITBL	FCB	\$6#,8
#123	WPITBL	FCB	\$#,8
#124	WPITBL	FCB	\$#,8
#125	WPITBL	FCB	\$#,8
#126	WPITBL	FCB	\$#,8
#127	WPITBL	FCB	\$#,8
#128	WPITBL	FCB	\$#,8
#129	WPITBL	FCB	\$#,8
#130	WPITBL	FCB	\$#,8
#131	GETSTAT	CMPS	\$55,READY
#132	GETSTAT	BNE	GETS#5
#133	GETSTAT	LDA	BUFIN,U
#134	GETSTAT	SUBA	BUFOUT,U
#135	GETSTAT	LBNE	GETEND
#136	GETSTAT	LDB	\$55,NOTRDY
#137	GETSTAT	LBR4	GETS#5
#138	GETSTAT	CMPS	\$55,EOF
#139	GETSTAT	LBSR	GETEND
#140	GETSTAT	CMPS	\$55,CURSR
#141	GETSTAT	LBSR	CURSTAT
#142	GETSTAT	CMPS	\$55,JOY
#143	GETSTAT	LBSR	JOYSTK
#144	GETSTAT	CMPS	\$131
#145	GETSTAT	BNE	GETS1#
#146	GETSTAT	LDA	\$57F
#147	GETSTAT	LBR4	GETS7#
#148	GETSTAT	CMPS	\$55,SCSIZ
#149	GETSTAT	BNE	GETS11
#150	GETSTAT	LDB	PD,RGS,Y
#151	GETSTAT	LDD	\$#
#152	GETSTAT	STD	R5X,X
#153	GETSTAT	LDD	\$24
#154	GETSTAT	STD	R5Y,X
#155	GETSTAT	CLRB	
#156	GETSTAT	RTS	ALL DONE
#157	GETSTAT		
#158	GETSTAT	CMPS	\$132
#159	GETSTAT	BNE	GETS2#
#160	GETSTAT	LDD	\$55#18
#161	GETSTAT	BRA	GETS7#
#162	GETSTAT		
#163	GETSTAT	CMPS	\$135
#164	GETSTAT	BNE	GETS3#
#165	GETSTAT	CLRA	
#166	GETSTAT	BRA	GETS7#
#167	GETSTAT	CMPS	\$15#
#168	GETSTAT	BREQ	GETS4#
#169	GETSTAT	CMPS	\$151
#170	GETSTAT	BREQ	GETS5#
#171	GETSTAT	CMPS	\$152
#172	GETSTAT	BNE	GETS6#
#173	GETSTAT	LEAY	CTRBL,PCR
#174	GETSTAT	TFR	X,D
#175	GETSTAT	TFR	Y,X
#176	GETSTAT	CMPS	\$55
#177	GETSTAT	BHI	GETS8#
#178	GETSTAT	CMPS	\$#
#179	GETSTAT	BLO	GETS8#
#180	GETSTAT	ABX	
#181	GETSTAT	LDA	,X
#182	GETSTAT	BRA	GETS7#
#183	GETSTAT		
#184	GETSTAT	LEAY	KEYTBL,PCR
#185	GETSTAT	GETS32	GET UNSHIFTED TABLE
#186	GETSTAT		GET VALUE OF KEY
#187	GETSTAT	LEAY	SHFTBL,PCR
#188	GETSTAT	BRA	GETS32
#189	GETSTAT		
#190	GETSTAT	CMPS	\$134
#191	GETSTAT	BNE	GETS8#
#192	GETSTAT	LDA	CURLST,PCR
#193	GETSTAT	ASRA	
#194	GETSTAT	BCC	GETS61
#195	GETSTAT	CLRB	
#196	GETSTAT	BRA	GETS62
#197	GETSTAT	LDB	\$1
#198	GETSTAT	ANDA	\$11111111
#199	GETSTAT	ASRA	
#200	GETSTAT	ASRA	
#201	GETSTAT	ASRA	
#202	GETSTAT	CMPS	\$#
#203	GETSTAT	BNE	GETS63
#204	GETSTAT	LDA	\$3

Continued on Page 13

COCO CADILLAC

Continued from Page 12

```
00205 0150 200D      BRA    GETS70
00206 015A 0101      GETS63 CMPA   #1      IS IT INVISIBLE ?
00207 015C 2003      BNE    GETS64      ...NO
00208 015E 4F        CLRA
00209 015F 2006      BRA    GETS70
00210 0161 0103      GETS64 CMPA   #3
00211 0163 2002      BNE    GETS70      ...NO:DEFAULTS TO FAST BLINK
00212 0165 0001      LDA    #1
00213 0167 AE26      GETS70 LDX    PD.RGS,Y
00214 0169 ED01      STD    RSD,X
00215 016B 5F        CLRB
00216 016C 2004      BRA    GETEND
00217
00218 016E AE26      GETS75 LDX    PD.RGS,Y
00219 0170 A701      STA    RSA,X
00220 0172 5F        GETEND CLRB
00221 0173 39        RTS
00222
00223 0174 C6D0      GETS80 LDB    #ESUNKSV
00224 0176 1A01      GETS90 ORCC   #1      SET CARRY ERROR!!!!!!
00225 0178 39        RTS
00226
00227 *****
00228 * PUTSTAT
00229 * FUNCTION KEY REPLACEMENT OF VALUES
00230 * IS THE ONLY ONE IMPLEMENTED
00231 *
00232 * A = CODE NO.
00233 * Y = ADDRESS OF PATH DESCRIPTOR
00234 * U = ADDRESS OF DEVICE STATIC STORAGE
00235 *
00236 * FOR FUNCKY
00237 *
00238 * B = NEW VALUE OF KEY
00239 * X = OFFSET OF KEY INTO TABLE
00240 *
00241
00242 0179 0196      PUTSTAT CMPA   #150   IS IT A FUNCTION KEY CHANGE ?
00243 017B 2721      BEQ    PUTS40   FOR UNSHIFTED KEYS
00244 017D 0197      CMPA   #151
00245 017F 2723      BEQ    PUTS50   FOR SHIFTED KEYS
00246 0181 0198      CMPA   #152
00247 0183 26EF      BNE    GETS80   NO:BAD CALL...
00248 0185 318D0240 PUTS09 LEAY   CRTBL,PCR GET CONTROL KEY TABLE
00249 0189 C137      PUTS10 CMPB   #55
00250 018B 22E7      BHI    GETS80   PAST TABLE END ?
00251 018D C100      CMPB   #0
00252 018F 25E3      BLO    GETS80   YES:SKIP(>55)
00253 0191 3404      PSHS   B          SAVE VALUE
00254 0193 1F10      TFR    X,D        PUT OFFSET INTO B
00255 0195 3502      PULS   A
00256 0197 1F21      TFR    Y,X        X POINTS TO TABLE
00257 0199 3A        ABX
00258 019A A704      STA    ,X        ADD IN KEY LOCATION
00259 019C 5F        CLRB
00260 019D 39        RTS
00261
00262 019E 318D01B7 PUTS40 LEAY   KEYTBL,PCR GET UNSHIFTED KEY TABLE
00263 01A2 20E5      BRA    PUTS10   STORE VALUE
00264
00265 01A4 318D01E9 PUTS50 LEAY   SHFTBL,PCR GET SHIFTED KEY TABLE
00266 01A8 20DF      BRA    PUTS10   STORE VALUE
00267
00268 *****
00269 * TERMINATION ROUTINE
00270 *
00271 * SINCE THIS ROUTINE IS NEVER REALLY
00272 * PERFORMED IT IS NOT NECESSARY
00273 * BUT IS HERE JUST IN CASE
00274 *
00275
00276 01AA 3401      TERM    PSHS   CC          SET EVERYTHING TO NORMAL
00277 01AC 1A10      ORCC   #0
00278 01AE DC6B      LDD    D.ALTRQ
00279 01B0 DD32      STD    D.IRQ
00280 01B2 3501      PULS   PC,CC
00281
00282 *****
00283 * READ ROUTINES
00284 *
00285 * Y = PATH DESCRIPTOR
00286 * U=STATIC STORAGE
00287 * RETURNS CHAR IN A
00288 * OR ERROR IN B, CC SET ON ERROR
00289 *
00290
00291
00292 01B4 30C834      READ    LEAX   INPBUF,U GET THE LOCATION OF THE INPUT
00293 01B7 E6C827      LDB    BUFOUT,U GET LAST CHAR OUT
00294 01BA 1A50      ORCC   #0
00295 01BC E1C826      CMPB   BUFIN,U  IF SAME AS LAST IN POSITION
00296 01BF 270F      BEQ    READ10   TELL CALLER AND WAIT
00297 01C1 3A        ABX
00298 01C2 A604      LDA    ,X        ELSE
00299 01C4 5C        INCB
00300 01C5 C17F      CMPB   #INPSI2-1 GET THE NEXT CHARACTER
00301 01C7 2301      BLS    READ5
00302 01C9 5F        CLRB
00303 01CA E7C827      READ5  STB    BUFOUT,U SAVE OUTPUT POINTER
00304 01CD 1CAE      ANDCC   #0
00305 01CF 39        RTS
00306
00307 01D0 A644      READ10 LDA    V.BUSY,U TELL CALLER
00308 01D2 A745      STA    V.WAKE,U THE ROUTINE BUSY
00309 01D4 1CAF      ANDCC   #0
00310 01D6 0E0000      LDX    #0
00311 01D9 103F0A      OS9    FSSLEEP   SLEEP FOREVER
00312 01DC 6F45      CLR    V.WAKE,U
00313 01DE 9E4B      LDX    $004B     GET PROC DESCRIPTOR
00314 01E0 E68036      LDB    $16,X
00315 01E3 27CF      BEQ    READ      TRY TO READ AGAIN
00316 01E5 C103      CMPB   #0
00317 01E7 22CB      BHI    READ      TRY READS FOR A WHILE
00318 01E9 43      COMA
00319 01EA 39        RTS
00320
00321 *****
00322 * KEYBOARD POLLING ROUTINES
00323 *
00324
00325
00326
00327 W 01EB FE006D      POLKEY LDU    >D.KBDSTA GET STATIC STORAGE
00328 01EE AEC81D      LDX    KPORT,U   GET PORT ADDRESS
00329 01F1 A603      LDA    3,X
00330 01F3 2004      BMI    POL10     ANY DATA?
00331                                YES: LOOK FOR IT
```

```
00331 W 01F5 6E9F0030      JMP    [D.SVCIRQ] GO AND SERVICE OTHER IRQ
00332 01F9 A602      POL10 LDA    2,X      CLEAR SEND PORT
00333 01FB 06FF      LDA    $FFF     SCAN FOR JOYSTICKS
00334 01FD A702      STA    2,X
00335 01FF A604      LDA    ,X
00336 0201 43      COMA
00337 0202 8403      ANDA   #003     REVERSE BITS
00338 0204 2609      BNE    JIGNOR   MASK OUT ALL BUT FIRES
00339 0206 6F02      CLR    2,X      IGNORE IF FIRE BUTTON DOWN
00340 0208 A604      LDA    ,X      ELSE LOOK AGAIN
00341 020A 43      COMA
00342 020B 847F      ANDA   #07F     REVERSE BITS
00343 020D 2625      BNE    GETKEY   MASK JOYSTICK ANALOG IN
00344 020F 06FF      JIGNOR LDA    $FFF     GET THE KEY THATS DOWN
00345 0211 A7C820      STA    KEY,U    NO KEYS ARE DOWN
00346 0214 A7C822      STA    KEYVAL,U CLEAR STORAGE
00347 0217 06FD      LDA    $FD      BUFFERS
00348 0219 A7C821      STA    LASTKY,U
00349 021C CC0503      JIG10 LDD    $0503  RESTORE REPEAT KEY VALUES
00350 021F EDC828      JIG20 STD    REPSFD,U
00351 W 0222 B6006F      JIG20 LDA    >D.DSKTMR ADJUST FOR MOTOR TIMER
00352 0225 2709      BEQ    JIGEND   FOR CLOCK ADJUSTMENT
00353 0227 4A      DECA
00354 W 0228 B7006F      STA    >D.DSKTMR
00355 022B 2603      BNE    JIGEND
00356 W 022D B7FF40      STA    $FF40
00357 W 0230 6E9F006B      JIGEND JMP    [D.ALTRQ]
00358 0234 8D53      BSR    FINDK    STROBE FOR KEY PRESSED
00359 0236 01FF      CMPA   $FFF     CHECK FOR SHIFT LOCK
00360 0238 27E8      BEQ    JIG20    IF NEGATIVE IGNORE...
00361 023A 0100      CMPA   #0
00362 023C 27DE      BEQ    JIG10    IF ZERO IGNORE
00363 023E 01FE      SHLOCK CMPA   #FE     IS IT THE SHIFT LOCK ?
00364 0240 2605      BNE    STORE   NO THEN STORE THE CHAR
00365 0242 63C823      COM    CAPLOK,U YES:REVERSE LOCK CONDITION
00366 0245 20DB      BRA    JIG20    RETURN
00367 0247 E6C826      LDB    BUFIN,U  CHECK INPUT BUFFER
00368 024A 30C834      LEAX   INPBUF,U
00369 024D 3A        ABX
00370 024E 5C        INCB
00371 024F C17F      CMPB   #INPSI2-1 IF NOT FULL STORE THE KEY
00372 0251 2301      BLS    STOR05   AND INCREMENT THE POSITION
00373 0253 5F        CLRB          ELSE START BUFFER AT TOP
00374 0254 E1C827      STOR05 CMPB   BUFOUT,U SAVE NEW BUFFER POS
00375 0257 2703      BEQ    STOR10
00376 0259 E7C826      STB    BUFIN,U  SAVE INPUT POSITION
00377 025C A784      STA    ,X        SAVE CHAR
00378 025E 271C      BEQ    WAKE     WAKE CALLER IF CHAR=0
00379 0260 A14D      CMPA   V.PCHR,U WAKE CHARACTER ?
00380 0262 2608      BNE    STOR20   NO:SKIP
00381 0264 AE49      LOX    V.DEV2,U
00382 0266 2714      BEQ    WAKE     TELL CALLER TO PAUSE
00383 0268 A708      STA    V.PAUS,X
00384 026A 2010      BRA    WAKE
00385 026C C603      STOR20 LDB    $$INTRPT KEYBOARD INTERRUPT SIGNAL
00386 026E A14B      CMPA   V.INTR,U INTERRUPT ?
00387 0270 2706      BEQ    STOR30   YES:TELL LAST CALLER
00388 0272 C602      LDB    $$SABORT ABORT SIGNAL
00389 0274 A14C      CMPA   V.QUIT,U QUIT ?
00390 0276 2604      BNE    WAKE     NO
00391 0278 A643      STOR30 LDA    V.LPRC,U GET LAST PROCESS
00392 027A 2004      BRA    WAKE10   AND WAKE IT
00393 027C C601      WAKE   LDB    $$SWAKE WAKE SIGNAL
00394 027E A645      LDA    V.WAKE,U WAKE CALLER
00395 0280 2703      BEQ    WAKE20   NO CALLER TO WAKE
00396 0282 103F08      OS9    FSSEND    CALLER TO WAKE
00397 0285 6745      WAKE20 CLR    V.WAKE,U RESET CALLER ID
00398 0287 2099      BRA    JIG20    FINISH
00399
00400 0289 4F      FINDK  CLRA
00401 028A 5F      CLRB
00402 028B A7C81F      STA    COLCNT,U  START THE LOOK CLEAN
00403 028E EDC824      STD    SHFLAG,U  BY CLEARING OUT FLAGS ECT.
00404 0291 43      COMA
00405 0292 17C820      STA    KEY,U
00406 0295 4A      DECA
00407 0296 A702      STA    2,X      CLEAR LAST VALUE
00408 0298 A604      LDA    ,X      SEE IF THIS COLUMN IS ON
00409 029A 43      COMA
00410 029B 847F      ANDA   #07F     MASK OUT JOY STICKS
00411 029D 270C      BEQ    NEXCOL   NOT ON:GET THE NEXT COLUMN
00412 029F C6FF      LDB    $FF      START FROM SCRATCH
00413 02A1 5C      FINDK2 INCB
00414 02A2 44      LSRA
00415 02A3 2402      BCC    FINDK3   IF KEY IS NOT FOUND
00416 02A5 0D00      BSR    MAKCHR   IF KEY IS FOUND
00417 02A7 C106      FINDK3 CMPB   #006     MORE ROWS?
00418 02A9 25F6      BCS    FINDK2   YES :LOOK MORE
00419 02AB 6CC81F      INC    COLCNT,U ALL ROWS EXHAUSTED
00420 02AE 1A01      ORCC   #0
00421 02B0 6902      ROL    2,X      SET CARRY
00422 02B2 25E4      BCS    FINDK1   MOVE TO NEXR COLUMN
00423 02B4 A6C820      GETVAL LDA    KEY,U   STILL NOT FOUND
00424 02B7 2B3A      BMI    NOKEY
00425 02B9 A1C821      CMPA   LASTKY,U
00426 02BC 277A      BEQ    REPEAT
00427 02BE A7C821      STA    LASTKY,U
00428 02C1 1F09      TFR    A,B
00429 02C3 3410      PSHS   X
00430 02C5 308D0090      LEAX   KEYTBL,PC
00431 02C9 3A        ABX
00432 02CA A6C824      LDA    SHFLAG,U
00433 02CD 2705      BEQ    GETV10
00434 02CF C638      LDB    $56
00435 02D1 3A        ABX
00436 02D2 2008      BRA    GETV20
00437 02D4 A6C825      GETV10 LDA    CTRFLG,U
00438 02D7 2703      BEQ    GETV20
00439 02D9 C670      LDB    #112
00440 02DB 3A        ABX
00441 02DC A604      GETV20 LDA    ,X
00442 02DE 0141      CMPA   #65
00443 02E0 250C      BLO    GETV30
00444 02E2 017A      CMPA   #122
00445 02E4 2208      BHI    GETV30
00446 02E6 015B      CMPA   #91
00447 02E8 253C      BLO    UPPERC
00448 02EA 0160      CMPA   #96
00449 02EC 2241      BHI    LOWERC
00450 02EE A7C822      GETV30 STA    KEYVAL,U
00451 02F1 3500      PULS   X,PC
00452
00453 02F3 06FD      NOKEY  LDA    $FD
00454 02F5 A7C821      STA    LASTKY,U
00455 02F8 6DC824      TST    SHFLAG,U
00456 02FB 2608      BNE    NOKEY9
00457 02FD 6DC825      TST    CTRFLG,U
00458 0300 2603      BNE    NOKEY9
00459 0302 06FF      LDA    $FFF
00460 0304 39        RTS
00461 0305 4F      NOKEY9 CLRA
00462 0306 39        RTS
00463
00464 0307 3404      MAKCHR PSHS   B          SAVE B
00465 0309 58      ASLB          MULTIPLY BY 8
00466 030A 58      ASLB
00467 030B 58      ASLB
```

Continued on Page 16

THE NEW COLOR COMPUTER

Continued from Page 1

by 24-line text display that is achieved by typing WIDTH 80. Add to that the eight background and foreground colors for the TEXT screen and a blinking cursor and underlining.

To best utilize the features of the Color Computer 3, Tandy has introduced a new RGB Analog monitor to take full advantage of the system hardware and provide 16 colors out of the palate of 64.

Other features include the documented 1.7 MHz clock; 32K ROM; 128K RAM (Standard, internally expandable to 512K); a 57-key keyboard including Control, Alternate, F1 and F2; text display capabilities of 32 x 16, 40 x 24 and 80 x 24; color graphics capabilities ranging from 64 x 32 (eight colors) to 640 x 192 (four colors) with six intermediate display formats including two displays that offer as much as 320 x 192 in 16 colors addressable through high resolution graphics, Extended BASIC and Program Paks.

I/O ports of the new machine include 1500 Baud cassette, two, 2-dimensional joysticks, RS-232C serial port, standard TV connector, composite video, RGB Analog and audio in addition to the standard ROM slot in the side.

As if this isn't enough, you have not heard the best news yet. This Color Computer lists for \$219.95! There's at least one vendor that is already listing this machine at the \$180 range.

When the Color Computer 3 is turned on, it has a few jobs to perform. It starts by jumping into the internal ROM at \$C000. This begins the process of copying the ROM to RAM and patching the ROM for the new commands. The new commands actually start at \$E000. Once completed, a short routine is copied into lower RAM where it then switches to the external ROM and checks for the presence of a DISK ROM and whether it is version 1.0 or 1.1 of RS-DOS.



OS-9 Users Group President Brian Lantz (left) and CompuServe SysOp Wayne Day discuss the new Tandy Color Computer at the press conference in New York.

Since the DOS must also be patched to access the new commands, this operation is both logical and practical. The problem comes when some other DOS is used, such as A-DOS or J-DOS. These enhanced operating systems tend to stretch past \$DFFF, which will overwrite the enhanced ECB commands. They also overwrite the check for DOS 1.0 and 1.1, so the Color Computer 3 is unable to determine the version of RS-DOS and defaults to none.

Some of the patches to ECB include the ability to LPEEK and LPOKE all 512K or RAM. This is the only area of BASIC that recognizes more than 128K, but may provide the information necessary for 512K spreadsheets in 80-columns.

Errors can now be trapped through the ON ERROR, ERNO and ERLIN commands. There's also an ON BREAK command.

The firebutton can now be recognized with the BUTTONN function,

removing the need to remember where to PEEK the firebutton.

The text screen is no longer limited to 32 x 16, which presents another problem. PRINT@ is not compatible in the 40 and 80-column modes. To fix this problem, the LOCATE(x,y) functions has been added to position the cursor on the screen. The remaining text screen commands are HSTAT (which gives the character, attributes and cursor information) and the ART command to set the text screen foreground and background colors, as well as the blinking and underline functions. All of the remaining Enhanced ECB commands are graphics oriented.

By this time next month, I hope to have had the time to run up my OS-9 Level II Color Computer and give you the run-down on what software runs and doesn't run. Until then . . . I think you'll love the newest addition to the Tandy line to home computers.

COMPUTERWARE® OS-9 Library

largest selection of OS-9 software anywhere!

Sourcerer

Menu driven symbolic 6809 disassembler produces symbolic source code that can be assembled with most editor-assemblers that use ASCII files.

CoCo \$39.95 SS-50 \$59.95

Advanced Utilities

KSHELL allows wildcards with any utility! Includes PATH for multiple path directories, UNLOAD for repetitive unlinks, FLINK for alternative boot files, ARCHIVE to backup large media to small media, & CPV.

CoCo \$29.95

Disk Fix & Utilities

Supports double-sided, double-density drives with 40 or 80 tracks & step rates to 6ms. Also includes DMODE, DIRCOPY, PATCH, FILELOOK, & COMPARE. Supports both 1.X & 2.X.

CoCo \$34.95

Textools

Utilities for manipulating text files.

CoCo \$29.95 SS-50 \$49.95

Look & Listen

Hires screen is 27% faster & saves over 3.5K. Sound utilities for sound without hardware & driver, descriptor, & command for Radio Shack SSC. Not compatible with OS-9 2.0

CoCo \$29.95

Business Applications

Comprehensive General Ledger, Check Ledger, Inventory, Accounts Payable, Accounts Receivable, & Payroll.

To Order

Add \$2 surface or \$5 air shipping. Calif residents add 6% sales tax.

RAID Debugger

A screen oriented debugger with built-in disassembler, single step, memory window, shell access, & more!

CoCo \$39.95

Macro Conditional Assembler

Supports all standard mnemonics & directives plus thrives on macros, conditional assembly, repeat sequences, & source library files.

CoCo \$49.95 SS-50 \$59.95

Advanced Editor

Full featured line editor for programming or text. Edits files larger than memory.

CoCo \$39.95 SS-50 \$49.95

Text Formatter

Interfaces with any editor's ASCII text files to produce beautiful documentation.

CoCo \$34.95 SS-50 \$54.95

The Last Word

The ultimate screen word processor for OS-9. Fast on-screen formatting, pop-down menus, Word-Pak compatible. Edits files larger than memory.

CoCo \$49.95

OS-9 BASIC

Full featured, easy-to-use, high level language. Supports graphics & joystick for CoCo. By invoking OS-9 commands you can write utilities!

CoCo \$49.95 SS-50 \$74.95

Ask for our free catalog!



COMPUTERWARE
Box 668
Encinitas, CA 92024
(619) 436-3512



In New York, Falsoft President Lonnie Falk (left) and RAINBOW Managing Editor Jim Reed (right) look at the new Color Computer with Tandy Buyer Barry Thompson.

HEY, ALL YOU OTHER GUYS

by George Dorner

Let's get this straight at the outset. I like my CoCo! I really like the MOTD. I like the UG Software Library. In fact, I like almost everything about OS-9... especially the growth of the OS-9 user community. But there is a tiny concern I have which goes back to ('scuse me while I stroke my beard) the days before Tandy embraced our favorite operating system and brought the broader exposure to OS-9 and the jump in membership for the UG.

Much of the appeal and much of the fun of the early (and tiny!) OS-9 community and Users Group were not due just to the knowledge that we were on to something good, something which was ahead of its time, something which was sophisticated and elegant. The breadth of the backgrounds of those involved and the scope of interests and applications of those using OS-9 intrigued most of us, I am sure. I have always been impressed at the great variety of folks I have met at the Microware seminars and even at the monthly meetings of our OS-9 group in Chicago. It's fascinating to rub elbows with users from major industrial firms, a major university, or with a bright individual with a good entrepreneurial idea which may use OS-9.

But I fret occasionally that, because most of the work being done for the UG just now is being done by users with at least one foot planted firmly in the CoCo world, some members or outsiders will get the feel that OS-9 is only for the CoCo. Worse, I worry that many OS-9 users may not experience the breadth and variety just alluded to and that this may limit or slow the growth of OS-9. That would be a shame, especially now that OS-9 is catching on in many quarters, and the number of applications is growing.

What to do to avoid "OS-9 myopia," the disease of only seeing one or two of the trees and missing the forest? Here's an antidote: Let's have more exposure for some of those OS-9 applications which most of us don't realize are going on. Let's hear about those VARs, OEMs, and OS-9 licensees who are outnumbered by the many CoCo users, SS-50 hobbyists, basement C hackers, who are so well represented here in the MOTD now. Let's hear the tales of how OS-9 is running on machines we didn't know existed, with software we would all give our left disk drive for, and in countries the names

of which we don't know how to pronounce. Tell us the stories of the research labs or corporate offices which use an OS-9 system innocuously and modestly outperforming a host of MS-DOS machines at some crucial tasks.

Well, one thing is sure. If nobody knows about such applications, nobody will write about them, and the myopia will continue. So, all you other guys, if you have an interesting or different OS-9 story to tell, dash off a few paragraphs and do yourself and the rest of us some good.

I know the stories are out there, and I will illustrate. I have talked to lots of OS-9ers in the past three years. I often end up talking long distance to some OS-9 zealot far away for long periods of time. One good member of the UG used to call me from various points around the country because he had no OS-9 contacts and had this lust to talk to someone... anyone!... about tree directories, pathnames, pipes, and whether OS-9 would ever have any graphics support. I still get those calls, and I regard that as a real bonus of my association with the OS-9 Users Group.

I've had a bunch of these contacts in the past week, and the breadth and variety of the OS-9 community really came through. One of these was a chat with a long-time acquaintance who is doing a development job on a system in which OS-9 will control 64 (count 'em!) 68000s. When this system is in production, you will probably be able to use one in your community... but without knowing that OS-9 is there. No more details until then.

I also talked to a developer who is putting together a 32-bit 68020 system on the VME buss with some really impressive performance benchmarks, 9-track tape support, and plans to market it as a business machine. These folks are hoping to beta test the 68020 version of OS-9 which has a new file manager for sequential blocks (SBFMAN?) and full memory management. Again, wait a couple of months for details.

A call came from Michigan from a CoCo hobbyist who had run his office on DynaStar and DynaCalc for over a year before installing a QT and moving over to Sculptor with great satisfaction both in the performance of the software and the hardware. His firm serves the auto industry, and while hearing a pitch from one of the biggies on a new

UNIX system he queried whether the UNIX system being quoted would outperform OS-9 in his application. The sales folks took a few moments to recall what OS-9 was, but then they acknowledged that OS-9 was probably faster for that application.

Finally, I just had the opportunity to use OS-9 on the IBM XT in our office! A friend called to ask if I would want to try out one of the TLM co-processor boards he had just purchased for their company. He didn't have to ask twice! We plugged it in and will report on our

findings next month. His firm is considering using it in a VAR application for real-time security monitoring, and they may place such a system with a customer so that a secretary can process words under MS-DOS while OS-9 is living in the same machine performing their applications. This is a real application of "If you can't beat 'em, ...".

All this OS-9 news in just a week! OS-9 is indeed alive, and well, and growing. I have been a little close-mouthed in describing the projects since all of them are still under development. But I hope everyone will write up details when the smoke clears and the projects are in place. We all will be a little richer and a lot prouder for knowing more about OS-9 being used in scientific, industrial, business, or whatever applications. (I would like to hear more about the Australian music synthesizer using OS-9.)

OS-9

LLOYD I/O INC.

Lloyd I/O is a computer engineering corporation providing software and hardware products and consulting services.

19535 NE GLISAN • PORTLAND, OR 97230 (USA)
PHONE: (503) 666-1097 • TELEX: 910 380 5448 LLOYD I/O

Computer Engineers

K-BASIC™ IS HERE

K-BASIC is a TSC XBASIC (XPC) compatible COMPILER for OS9 & FLEX... price \$199

Here at last is a compiler for BASIC that will compile all your XBASIC programs. K-BASIC compiles TSC's XBASIC and XPC programs to machine code. K-BASIC is ready now to save you money and time by teaching your computer to:
• Think Faster • Conserve Memory • Be Friendlier

Call (503) 666-1097 for our CATALOG.

We have many programs for serious software developers!

Micro BASIC for OS9... \$149

A structured micro BASIC for general system control featuring Parameter passing, 10 string variables, 26 numeric variables, subroutines, nested loops, interactive I/O, sequential files, and time variables (for applications executing in the background required to execute procedures such as disk or file backups). Includes the SEARCH and RESCUE UTILITIES™. (For OS9 ONLY)

SEARCH and RESCUE UTILITIES™

for OS9... \$35

A super directory search utility. Output may be piped to the included utilities to perform file: COPIES, DELETES, MOVES, LISTING (pagination), and FILTERING. Some filtering utility programs are included, of interest is the FILE DATE CHECKING utilities YOUNGER and DRAFT (Level 2). (For OS9 Level 1 and 2)

PATCH™

Modem Communications for OS9... \$39

PATCH is a modem communications program for OS9 featuring KEY MACROS, ASCII TEXT AND BINARY FILE UP/DOWN LOADING, PRINTER COPY, and HELP MENUS. We use it several times each day with our TELEX service. PATCH is convenient and easy to use. Key macros may be pre-stored and loaded at any time.

CRASMB™

CROSS ASSEMBLER PACKAGE

for OS9 & FLEX... all for \$399

Motrola CPU's... \$150
Intel CPU's... \$150. Others... \$150

CRASMB is the highly acclaimed cross assembler package for OS9 and FLEX systems. It turns your high level code into a development ready machine code. Get it now!

NOW FOR OS9 68K \$249

CRASMB features: Modules, Conditional Long symbol names, Symbol cross reference table, Object code in 4 formats (OS9, FLEX, S-1-S9, INTEL HEX).

VISA, MC, COD, CHECKS, ACCEPTED

USA: LLOYD I/O (503 666 1097), S.E. MEDIA (800 338 6800)
England: Vivaway (0582 423425), Windrush (0692 405189)
Germany: Zacher Computer (65 25 299), Kell Software (06203 6741)
Australia: Parts Radio Electronics (344 9111)
Japan: Microboards (0474) 22-1741, Seikou (03) 832-6000
Switzerland: Elsoft AG (056 86 27 24)
Sweden: Micromaster Scandinavian AG (018 - 138595)

K-BASIC, DQ, SEARCH and RESCUE UTILITIES,
PATCH, CRASMB, and CRASMB 16.32 are trademarks of LLOYD I/O
OS9 is a ™ of Microware. ILLX is a ™ of TSC

Continued from Page 13

00468	039C EBC81F		ADDB	COLCHT,U	ADD IN COLUMN
00469	039F C134		CHPB	0034	IS CHARACTER THE CTRL KEY ?
00470	0311 2685		BNE	MAKC10	NO CHECK FOR SHFT
00471	0313 6CC825		INC	CTRFLG,U	INCREMENT CONTROL KEY
00472	0316 3584		PULS	PC,0	RETURN
00473	0318 C137	MAKC10	CHPB	0037	IS IT SHFT?
00474	031A 2685		BNE	MAKC20	NO;GO STORE THE KEY
00475	031C 63C824		COM	SHFLAG,U	ELSE TURN SHFT FLAG
00476	031F 3584		PULS	PC,0	AND RETURN
00477	0321 E7C820	MAKC20	STB	KEY,U	
00478	0324 3584		PULS	PC,0	RETURN
00479					
00480					
00481	0326 6DC823	UPPERC	TST	CAPLOK,U	
00482	0329 27C3		BEQ	GETV30	
00483	032B 8820		ADDA	0020	
00484	032D 20B7		BRA	GETV30	
00485					
00486	032F 6DC823	LOWERC	TST	CAPLOK,U	
00487	0332 27BA		BEQ	GETV30	
00488	0334 8020		SUBA	0020	
00489	0336 20B6		BRA	GETV30	
00490					
00491	0338 A6C820	REPEAT	LDA	REPSPD,U	
00492	033B 2707		BEQ	REP10	
00493	033D 4A		DECA		
00494	033E A7C820		STA	REPSPD,U	
00495	0341 86FF		LDA	00FF	
00496	0343 39		RTS		
00497	0344 8605	REP10	LDA	0005	
00498	0346 A7C820		STA	REPSPD,U	
00499	0349 A6C829		LDA	REPWAT,U	
00500	034C 2707		BEQ	REP20	
00501	034E 4A		DECA		
00502	034F A7C829		STA	REPWAT,U	
00503	0352 86FF		LDA	00FF	
00504	0354 39		RTS		
00505	0355 A6C822	REP20	LDA	KEYVAL,U	
00506	0358 39		RTS		
00507					
00508					
00509					
00510					
00511					
00512					
00513					
00514	0359 40616263	KEYTBL	FCB	040,061,062,063	0,A,B,C
00515	035D 64656667		FCB	064,065,066,067	D,E,F,G
00516	0361 68696A6B		FCB	068,069,06A,06B	H,I,J,K
00517	0365 6C6D6E6F		FCB	06C,06D,06E,06F	L,M,N,O
00518	0369 70717273		FCB	070,071,072,073	P,Q,R,S
00519	036D 74757677		FCB	074,075,076,077	T,U,V,W
00520	0371 78797A0C		FCB	078,079,07A,08C	X,Y,Z,UA
00521	0375 0A0B0920		FCB	08A,080,089,020	DA,LA,RA,SP
00522	0379 30313233		FCB	030,031,032,033	0,1,2,3
00523	037D 34353637		FCB	034,035,036,037	4,5,6,7
00524	0381 38393A3B		FCB	038,039,03A,03B	8,9,1,1
00525	0385 3C3D3E3F		FCB	03C,03D,03E,03F	2,3,4,5
00526	0389 0D0E0F51B		FCB	08D,080,085,01B	EN,CL,BR,ALT
00527	038D 005C5F00		FCB	000,05C,05F,000	CTR,F1,F2,SH
00528					
00529	0391 00414243	SHFTBL	FCB	060,041,042,043	0,A,B,C
00530	0395 44454647		FCB	044,045,046,047	D,E,F,G
00531	0399 48494A4B		FCB	048,049,04A,04B	H,I,J,K
00532	039D 4C4D4E4F		FCB	04C,04D,04E,04F	L,M,N,O
00533	03A1 00515253		FCB	050,051,052,053	P,Q,R,S
00534	03A5 04555657		FCB	054,055,056,057	T,U,V,W
00535	03A9 08595A5C		FCB	058,059,05A,01C	X,Y,Z,UA
00536	03AD 1A101920		FCB	01A,010,019,020	DA,LA,RA,SP
00537	03B1 30212223		FCB	030,021,022,023	0,1,2,3
00538	03B5 24252627		FCB	024,025,026,027	4,5,6,7
00539	03B9 28292A2B		FCB	028,029,02A,02B	8,9,1,1
00540	03				

00595	0447	0100		CMPS	000	A BACKSPACE ?
00596	0449	10270164		LREQ	CURLFT	...YES
00597	044D	0100		CMPS	000	A MOVE CURSOR RIGHT ?
00598	0447	102700AF		LREQ	ADVCUR	...YES
00599	0453	0100		CMPS	000	A CURSOR UP ?
00600	0455	10270172		LREQ	CURUP	...YES
00601	0459	0100		CMPS	000	CIS AND HOME ?
00602	045B	102700E8		LREQ	HOMCLS	YES
00603	045F	0100		CMPS	000	ERASE ENTIRE CURSOR LINE ?
00604	0461	10270107		LREQ	ERLINE	...YES
00605	0465	0100		CMPS	000	ERASE TO END OF CURSOR LINE ?
00606	0467	10270101		LREQ	ERLINE	...YES
00607	046B	0100		CMPS	000	CLEAR TO END OF SCREEN ?
00608	046D	1027010F		LREQ	EXTEND	...YES
00609	0471	0100		CMPS	000	HOM - NO CIS
00610	0473	102700DD		LREQ	HOM	...YES
00611	0477	0100		CMPS	000	CHANGE CURSOR START ?
00612	0479	2714		BEQ	WRIT22	GO START ROUTINE
00613	047B	0100		CMPS	000	X,Y POSITIONING ?
00614	047D	2710		BEQ	WRIT22	...YES
00615	047F	0110		CMPS	000	ESCAPE CHAR
00616	0481	270C		BEQ	WRIT22	...YES
00617	0483	011D		CMPS	000	UNSHIFTED KEY CHANGE ?
00618	0485	2700		BEQ	WRIT22	...YES
00619	0487	011E		CMPS	000	SHIFTED KEY CHANGE ?
00620	0489	2704		BEQ	WRIT22	...YES
00621	048B	011F		CMPS	000	CHANGE FUNCTION KEY ?
00622	048D	2600		BNE	WRIT30	...NO; TRY BELL
00623	048F	A7C02B	WRIT22	STA	ESCFLG,U	STORE FOR FURTHER
00624	0492	5F		CLRB		ACTION AND SET
00625	0493	E7C02A		STB	ESCONT,U	THE ESC FLAG
00626	0496	39		RTS		DONE
00627	0497	ALA033	WRIT30	CMPS	PD.OVF,Y	IS IT A BELL ?
00628	049A	10270323		LREQ	BELL	...YES;GO RING IT
00629	049E	5F		CLRB		ELSE JUST
00630	049F	39		RTS		RETURN
00631						
00632						*****
00633						* PRINT A CHARACTER AND ADVANCE
00634						* A=CHARACTER TO PRINT
00635						
00636	04A0	0D04	PRTCHR	BSR	PRTCHR	PUT THE CHAR ON SCR
00637	04A2	0D5E		BSR	ADVCUR	AND ADVANCE THE CURSOR
00638	04A4	5F		CLRB		RETURN WITH NO ERR
00639	04A5	39		RTS		
00640						
00641	04A6	6DC02F	PRTCHR	TST	INVFLG,U	IS IT INVERSE VIDEO
00642	04A9	2704		BEQ	PUTC10	NO;SKIP
00643	04AB	0A00		ORA	0100000000	...YES; SET
00644	04AD	2002		BRA	PUTC11	SKIP
00645	04AF	047F	PUTC10	ANDA	0001111111	MAKE NORMAL
00646	04B1	0D00	PUTC11	BSR	CHKPRT	MAKE SURE PORT IS READY
00647	04B3	3410		PSHS	X	SAVE X
00648	04B5	AE41		LDX	1,U	GET PORT ADDRESS
00649	04B7	A703		STA	03,X	SEND CHAR
00650	04B9	3510		PULS	X	RESTORE X
00651	04BB	39		RTS		GO BACK
00652						
00653	04BC	C61F	CHKPRT	LDB	001F	IS THE
00654	04BE	E7D001		STB	[1,U]	LATCH READY
00655	04C1	6D0001	CHKP10	TST	[1,U]	IF NOT
00656	04C4	2AFB		BPL	CHKP10	JUST WAIT TILL IT IS
00657	04C6	39		RTS		DONE
00658						
00659	04C7	0DF3	GETCHA	BSR	CHKPRT	IS THE PORT READY
00660	04C9	3410		PSHS	X	SAVE X
00661	04CB	AE41		LDX	1,U	AND STORE
00662	04CD	A003		LDA	03,X	THE CHAR
00663	04CF	3510		PULS	X	RESTORE X
00664	04D1	39		RTS		RETURN
00665						
00666						*****
00667						* CLEAR A PART

Continued on Page 17

COCO CADILLAC

Continued from Page 16

```
00732 0544 8DA8 BSR UPDWP IN CRTX
00733 0546 39 RTS
00734
00735 0547 8E0000 HOMCLS LDX #0 HOME AND CLEAR THE SCREEN
00736 054A A7C82F STX DSTART,U ZERO SCREEN START
00737 054D 108E0000 LDY #0000 LENGTH OF SCREEN
00738 0551 17FF7E LBSR CLEAR CLEAR IT
00739 0554 4F HOME CLRA SET POSITION TO
00740 0555 A7C82D STA CURLIN,U TOP LINE
00741 0558 A7C82E STA CURCOL,U COLUMN #
00742 055B A7C83F LDX DSTART,U SET CRTX
00743 055E C60C LDB #0C NEW TOP
00744 0560 8D8C BSR UPDWP SET POSTION
00745 0562 C612 LDB #012
00746 0564 8D88 BSR UPDWP
00747 0566 C60E LDB #00E AND CURSOR
00748 0568 8D84 BSR UPDWP
00749 056A 5F CLRB NO ERRORS
00750 056B 39 RTS RETURN
00751
00752 056C 8103 ERLINE CMPA #1 IS IT ERASE WHOLE LINE?
00753 056E 2603 BNE ERLI10 NO;
00754 0570 6FC82E CLR CURCOL,U YES; START AT THE BEGINNING
00755 0573 C600 LDB #0 CLEAR TO END
00756 0575 E0C82E SUBB CURCOL,U COUNT IS NOW IN B
00757 0578 4F ERLI10 CLRA
00758 0579 1F02 TFR D,Y PUT COUNT INTO Y
00759 057B 8D24 BSR FINDPO GET THE POSITION
00760 057D 16FF52 LBRA CLEAR CLEAR THE SCREEN
00761
00762 0580 8D1F ERTEND BSR FINDPO GET POSITION
00763 0582 A7C832 STX WORK,U SAVE IT
00764 0585 E0C83F LDO DSTART,U FIND SCREEN END
00765 0588 C30780 ADDD #0780
00766 058B A7C832 SUBD WORK,U ADJUST CLEAR COUNT
00767 058E 1F02 TFR D,Y PUT COUNT IN Y
00768 0590 16FF3F LBRA CLEAR CLEAR THE SCREEN
00769
00770 0593 8D0C MOVCUR BSR FINDPO MOVE THE CURSOR
00771 0595 C60E LDB #00E TO THE CURRENT POSITION
00772 0597 17FF54 LBSR UPDWP
00773 059A C612 LDB #012 SET POSITION
00774 059C 17FF4F LBSR UPDWP
00775 059F 5F CLRB
00776 05A0 39 RTS
00777
00778 05A1 A6C82D FINDPO LDA CURLIN,U GET LINE #
00779 05A4 C600 LDB #0 AND FIGURE THE POS
00780 05A6 3D MUL IN SCREEN
00781 05A7 E3C83F ADDD DSTART,U MEMORY
00782 05AA 1F01 TFR D,X PUT IT IN X
00783 05AC E6C82E LDB ADD IN THE COL
00784 05AF 3A ABX COUNT TOO
00785 05B0 39 RTS RETURN
00786
00787 05B1 A6C82E CURLFT LDA CURCOL,U BACKSPACE
00788 05B4 4D TSTA ADD START OF LINE?
00789 05B5 260C BNE CURLI0 NO
00790 05B7 8650 LDA #0 YES; MOVE TO END
00791 05B9 E6C82D LDB CURLIN,U LEFT ONE POSITION
00792 05BC 5D TSTB AT TOP ?
00793 05BD 278A BEQ CURL90 YES;SKIP LINE CHANGE
00794 05BF 5A DECB GO UP ONE LINE
00795 05C0 E7C82D STB CURLIN,U SAVE IT
00796 05C3 4A DECA GO BACK ONE CHARACTER
00797 05C4 A7C82E STA CURCOL,U SAVE NEW COLUMN
00798 05C7 28CA BRA MOVCUR MOVE THE CURSOR
00799 05C9 5F CLRB
00800 05CA 39 RTS
00801
00802 05CB 6DC82D CURUP TST CURLIN,U MOVE UP ONE LINE
00803 05CE 27F9 BEQ CURL90 ALREADY AT TOP SKIP
00804 05D0 6AC82D DEC CURLIN,U MOVE UP
00805 05D3 28BE BRA MOVCUR
00806
00807 05D5 E6C82D CURDWN LDB CURLIN,U LINEFEED
00808 05D8 5C INCB
00809 05D9 C118 CMPB #24 PAST SCREEN END ?
00810 05DB 2B05 BMI CURD10 NO;IGNORE
00811 05DD 17FF43 LBSR SCROLL ELSE SCROLL
00812 05E0 2003 BRA CURD20 LINE # SAME
00813 05E2 E7C82D CURD10 STB CURLIN,U SAVE NEW LINE #
00814 05E5 20AC CURD20 BRA MOVCUR MOVE THE CURSOR
00815
00816 05E7 6DC82A ESCAPE TST ESCONT,U TEST COUNT
00817 05EA 2658 BNE DOESC GO IF NOT 0
00818 05EC 8141 CMPA #041 IS IT A CLEAR TO END
00819 05EE 2605 BNE ESCA01 NO;SKIP
00820 05F0 8604 LDA #04 YES TRANSATE AND EXECUTE
00821 05F2 16FF77 LBRA ERLINE
00822 05F5 8142 CMPA #042 IS IT A CLEAR TO END
00823 05F7 2787 BEQ ERTEND YES;GO DO IT
00824 05F9 8156 CMPA #056 IS IT A BLOCK CURSOR
00825 05FB 2610 BNE ESCA10 NO;SKIP
00826 05FD A68DFA5 LDA CURLST,PCR CHANGE CURSOR
00827 0601 84F8 ANDA #011111000
00828 0603 1F01 TFR D,X UPDATE CURSOR
00829 0605 C60A LDB #010 TYPE REG
00830 0607 17FEE4 LBSR UPDWP
00831 060A 108109 LBRA CTREND CLEAR ESCAPE FLAG
00832 060D 8176 CMPA #076 UNDERLINE CURSOR ?
00833 060F 2608 BNE ESCA20 NO;SKIP
00834 0611 EC8DFAA1 LDD CURLST,PCR CHANGE CURSOR
00835 0615 8A07 ORA #000001111
00836 0617 20EA BRA ESCA05
00837 0619 8153 ESCA20 CMPA #053 SET INVERS/NORMAL?
00838 061B 2608 BNE ESCA30 NO;SKIP
00839 061D 6CC82A ESCA21 INC ESCONT,U YES WAIT FOR SECOND CHAR
00840 0620 A7C82C STA ESCBUF,U STORE CODE FOR NEXT PROCESSI
00841 0623 5F CLRB
00842 0624 39 RTS RETURN NO ERRORS
00843 0625 8157 ESCA30 CMPA #057 SET CURSOR RATE?
00844 0627 27F4 BEQ ESCA21 YES;WAIT FOR NEXT CHAR
00845 0629 8145 CMPA #045 INSERT A LINE
00846 062B 102701CB LBEQ DINLIN GO DO IT
00847 062F 8146 CMPA #046 DELETE A LINE
00848 0631 102701C5 LBEQ DINLIN GO DO IT
00849 0635 8147 CMPA #047 SHIFT LINE RIGHT 8
00850 0637 102701AE LBEQ SHFLIN YES
00851 063B 8148 CMPA #048 SHIFT LINE LEFT 8
00852 063D 102701A8 LBEQ SHFLIN YES;
00853 0641 1600D2 LBRA CTREND ELSE POOR TRY
00854
00855 0644 E6C82C DOESC LDB ESCBUF,U FIND OUT WHAT'S WAITING
00856 0647 C153 CMPB #053 IS IT A REVERSE VIDEO ?
00857 0649 2619 BNE DOES10 NO;SKIP
00858 064B 8120 CMPA #020 NORMAL MODE ?
```

```
00859 064D 2607 BNE DOES01 NO SKIP
00860 064F 4F CLRA ELSE SET FLAG FOR NORMAL
00861 0650 A7C82F STA INVFLG,U AND SAVE IT
00862 0653 1600C0 LBRA CTREND CLEAR ESCAPE
00863 0656 8121 DOES01 CMPA #021 IS IT INVERSE?
00864 0658 102600BA LBNE CTREND NO;BAD CALL-CLEAR IT
00865 065C 867F LDA #011111111 ELSE SET FOR INVERSE
00866 065E A7C82F STA INVFLG,U SAVE IN FLAG
00867 0661 1600B2 LBRA CTREND CLEAR ESCAPE
00868 0664 C157 DOES10 CMPB #057 IS IT A CHANGE IN CURSOR RAT
00869 0666 102600AC LBNE CTREND NO;CLEAR ESCAPE
00870 066A 8120 CMPA #020 STEADY CURSOR?
00871 066C 2609 BNE DOES11 NO;TRY NEXT
00872 066E A68DFA44 LDA CURLST,PCR CHANGE CURSOR
00873 0672 840F ANDA #000001111
00874 0674 160092 LBRA CURS10 AND STORE IT
00875 0677 8122 DOES11 CMPA #022 FAST BLINK?
00876 0679 2608 BNE DOES12 NO...
00877 067B EC8DFA37 LDD CURLST,PCR MAKE A FAST BLINK
00878 067F 840F ANDA #000001111
00879 0681 8B40 ADDA #040
00880 0683 160003 LBRA CURS10
00881 0686 8122 DOES12 CMPA #022 INVISIBLE ?
00882 0688 260A BNE DOES13 NO
00883 068A EC8DFA28 LDD CURLST,PCR MAKE INVISIBLE
00884 068E 840F ANDA #000001111
00885 0690 8B20 ADDA #020
00886 0692 2075 BRA CURS10
00887 0694 EC8DFA1E DOES13 LDD CURLST,PCR DEFAULT IS A SLOW BLINK
00888 0698 840F ANDA #000001111
00889 069A 8B60 ADDA #060
00890 069C 2068 BRA CURS10
00891
00892 069E E6C82A FUNKEY LDB ESCONT,U ALL CHARACTERS READ
00893 06A1 C101 CMPB #1
00894 06A3 2709 BEQ FUNK10 YES; CHANGE KEY
00895 06A5 5C INCB NO; INCR COUNT
00896 06A6 E7C82A STB ESCONT,U AND STORE IT
00897 06A9 A7C82C STA ESCBUF,U AND CODE
00898 06AC 5F CLRB RETURN NO ERROR
00899 06AD 39 RTS
00900 06AE 1709 FUNK10 TFR A,B YES PUT OFFSET IN B
00901 06B0 4F CLRA
00902 06B1 1F01 TFR D,X
00903 06B3 A6C82B LDA ESCFLG,U PUT NEW CODE INTO A
00904 06B6 811D CMPA #01D UNSHIFTED KEY CHANGE ?
00905 06B8 270C BEQ FUNK20 ...YES
00906 06BA 811E CMPA #01E SHIFTED KEY ?
00907 06BC 2710 BEQ FUNK30 ...YES
00908 06BE E6C82C LDB ESCBUF,U
00909 06C1 17FAC1 LBSR PUTS09 AND CHANGE KEY
00910 06C4 2050 BRA CTREND CLEAR CONTROL CALL
00911
00912 06C6 E6C82C FUNK20 LDB ESCBUF,U
00913 06C9 17FAD2 LBSR PUTS40 SAVE UNSHIFTED VALUE
00914 06CC 2048 BRA CTREND
00915
00916 06CE E6C82C FUNK30 LDB ESCBUF,U
00917 06D1 17FAD0 LBSR PUTS50 SAVE SHIFTED VALUE
00918 06D4 2048 BRA CTREND
00919
00920 06D6 E6C82A XYPOS LDB ESCONT,U GET COUNT
00921 06D9 C101 CMPB #1 ALL VALUES GATHERED ?
00922 06DB 2709 BEQ XYPOS1 YES;MOVE THE CURSOR
00923 06DD A7C82C STA ESCBUF,U NO; SAVE VALUE
00924 06E0 5C INCB INCREMENT COUNT
00925 06E1 E7C82A STB ESCONT,U AND STORE
00926 06E4 5F CLRB RETURN NO ERRORS
00927 06E5 39 RTS
00928 06E6 0020 XYPOS1 SUBA #020 ADJUST VALUE
00929 06E8 A7C82D STA CURLIN,U STORE Y
00930 06EB A6C82C LDA ESCBUF,U GET X
00931 06EE 0020 SUBA #020 ADJUST
00932 06F0 A7C82E STA CURCOL,U STORE
00933 06F3 17FE9D LBSR MOVCUR MOVE THE CURSOR
00934 06F6 201E BRA CTREND
00935
00936 *****
00937 * INSTEAD OF COLORS FOR 05 NN
00938 * THIS CHANGES SHAPES AND BLINKS
00939 * ONLY ALTERNATIVE THAT DOES SOMETHING
00940
00941 06F8 0020 CURSOR SUBA #020 STRIP OFFSET
00942 06FA 2B1A BMI CTREND IF NOT IN RANGE
00943 06FC 8109 CMPA #9
00944 06FE 2216 BMI CTREND IGNORE COMMAND
00945 0700 1709 TFR A,B PUT INTO B
00946 0702 308D0018 LEAX CURTBL,PCR TO FIND TYPE
00947 0706 3A ABX IN TABLE
00948 0707 A684 LDA ,X GET CURSOR MSB
00949 0709 C608 LDB #8 GET NORMAL LSB
00950 070B 1F01 TFR D,X PUT IT IN X
00951 070D AF8DF9A5 STX CURLST,PCR SAVE NEW CURSOR
00952 0711 C60A LDB #00A UPDATE CURSOR REG
00953 0713 17FDD8 LBSR UPDWP IN CRTX
00954 0716 5F CLRB CLEAR ALL ESCAPE FLAGS
00955 0717 E7C82B STB ESCFLG,U
00956 071A E7C82A STB ESCONT,U AND COUNTERS
00957 071D 39 RTS AND RETURN
00958
00959 071E 606700 CURTBL FCB $60,$67,$ SLOW BLOCK,SLOW UL,STEADY BL
00960 0721 404707 FCB $40,$47,$ FAST BLK, FAST UL,STDY UL
00961 0724 6444 FCB $64,$44 SLOW HALF BLK,FAST 1/2 BLK
00962 0726 6646 FCB $66,$46 SLOW 3/4 BLK,FAST 3/4 BLOCK
00963
00964 *****
00965 * JOYSTICK STATUS
00966 *
00967 * X = 0 FOR RIGHT STICK
00968 * 1 FOR LEFT
00969 *
00970 * RETURNS :
00971 * X = X VALUE
00972 * Y = Y VALUE
00973 * A = 0 FOR FIRE OFF,$FF FOR ON
00974 *
00975
00976 0728 AE26 JOYSTK LDX PD.RGS,Y CHECK CALLER'S REGS
00977 072A 3421 PSRS Y,CC SAVE CC AND DESCRIPTOR ADDRE
00978 072C 1A10 ORCC #010 DISABLE INTERRUPTS
00979 072E 86FF LDA #FFF CHECK FOR FIRE
00980 W 0730 7FFF02 CLR $FFF
00981 W 0733 F6FF00 LDB $FFF
00982 0736 10AE04 LDY R0X,X LEFT STICK
00983 0739 2606 BNE JOYS10 YES;GO
00984 073B C401 ANDB #01 SET FOR RIGHT FIRE
00985 073D 2606 BNE JOYS20 ON
00986 073F 2005 BRA JOYS21 NOT ON
00987 0741 C402 JOYS10 ANDB #002
00988 0743 2701 BEQ JOYS21
00989 0745 4F JOYS20 CLRA NO FIRE
00990 0746 A701 JOYS21 STA R0A,X STORE FIRE
00991 W 0748 B6FF03 LDA $FFF GET X
00992 074B 8A08 ORA #008 POSITION
00993 074D 10AE04 LDY R0X,X GET OLD X VALUE
00994 0750 2602 BNE JOYS30 IF RIGHT
00995 0752 8477 ANDA #0F7 ELSE MASK
```

Continued on Page 18

COCO CADILLAC

Continued from Page 17

```
00996 W 0754 B7FF03 JOYS30 STA $FF03 SET UP TO READ STICK
00997 W 0757 B6FF01 LDA $FF01
00998 075A 84F7 ANDA $F7
00999 075C 8D15 BSR JOYXY GO GET VALUE
01000 075E ED04 STD RSX,X SAVE VALUE RETURNED
01001 W 0760 B6FF01 LDA $FF01 AND READ
01002 0763 8A00 ORA $00
01003 0765 8D0C BSR JOYXY AGAIN FOR Y VALUE
01004 0767 3406 PSHS B,A
01005 0769 CC003F LDD $0003F REMOVE OFFSET
01006 076C A3E1 SUBD ,S++
01007 076E ED06 STD RSX,X SAVE IT
01008 0770 5F CLRB NO ERROR
01009 0771 35A1 PULS PC,Y,CC RESTORE AND RETURN
01010
01011 W 0773 B7FF01 JOYXY STA $FF01 READ STICK X OR Y
01012 0776 5F CLRB POSTION
01013 0777 8D0A BSR JXY10 MAKE VALUE
01014 0779 8D08 BSR JXY10
01015 077B 8D06 BSR JXY10
01016 077D 8D04 BSR JXY10
01017 077F 54 LSRB
01018 0780 54 LSRB
01019 0781 4F CLRA
01020 0782 39 RTS
01021
01022 0783 3404 JXY10 PSHS B SAVE VALUE IN B
01023 0785 867F LDA $07F PUT $7F INTO
01024 0787 1F09 TFR A,B A AND B
01025 0789 54 LSRB BUILD UP JOYSTICK VALUE
01026 078A C103 CMPB $003
01027 078C 2207 BHI JXY20 GO CHECK NEXT DAC INPUT
01028 078E 44 LSRA DIVIDE BY 2 TWICE
01029 078F 44 LSRA
01030 0790 1F09 TFR A,B MAKE VALUE
01031 0792 EBE0 ADDB ,S+
01032 0794 39 RTS RETURN
01033
01034 0795 CB02 JXY20 ADDB $002 READ EACH VALUE
01035 0797 C4FC ANDB $0FC
01036 0799 3404 PSHS B
01037 W 079B B7FF20 STA $FF20
01038 W 079E 7DFF00 TST $FF00
01039 07A1 2A04 BPL JXY21 IF POSITIVE GO
01040 07A3 ABE0 ADDA ,S+ OTHERWISE ADD VALUE
01041 07A5 20E2 BRA JXY11
01042 07A7 ABE0 JXY21 SUBA ,S+ SUBTRACT VALUE
01043 07A9 20DE BRA JXY11
01044
```

```
01045 *****
01046 * CURSOR POSITION - GETSTAT
01047 *
01048
01049 07AB AE26 CURSTAT LDX PD.RGS,Y TELL CALLER X,Y
01050 07AD E6C82D LDB CURLIN,U POSITIONS
01051 07B0 4F CLRA
01052 07B1 ED06 STD RSX,X Y INTO REGS Y
01053 07B3 E6C82E LDB CURCOL,U
01054 07B6 ED04 STD RSX,X X INTO X
01055 07B8 17FD0C LBSR GETCHA CHARACTER UNDER THE CURSOR
01056 07BB E701 STB R0A,X INTO A
01057 07BD 1F08 TFR B,A
01058 07BF 5F CLRB NO ERRORS
01059 07C0 39 RTS
01060
01061 07C1 108E012C BELL LDY $300 RING THE BELL;TIME COUNT
01062 W 07C5 B6FF22 BELL10 LDA $FF22 USING SINGLE BIT SOUND
01063 07C8 8A02 ORA $00000010 SET UP FOR HIGH PART
01064 W 07CA B7FF22 STA $FF22 OF SQUARE WAVE
01065 07CD 8E0014 LDX $20 DURATION FOR PITCH
01066 07D0 301F BELL20 LEAX -1,X COUNT IT DOWN
01067 07D2 26FC BNE BELL20
01068 W 07D4 B6FF22 LDA $FF22 THEN SOUND LOW PART
01069 07D7 84FD ANDA $01111101 OF WAVE
01070 W 07D9 B7FF22 STA $FF22
01071 07DC 8E0014 LDX $20 COUNT AGAIN
01072 07DF 301F BELL30 LEAX -1,X UNTILL DONE
01073 07E1 26FC BNE BELL30
01074 07E3 313F LEAY -1,Y RUNG ENOUGH?
01075 07E5 26DE BNE BELL10 NO;RING SOME MORE
01076 07E7 5F CLRB NO ERRORS
01077 07E8 39 RTS
01078
01079 *****
01080 * ROUTINES FOR OPAK COMPATIBILITY
01081 *
01082 *
01083
01084 07E9 3402 SHFLIN PSHS A SHIFT LINE 8 POSITIONS
01085 07EB 8D23 BSR LINSTR GET START OF LINE
01086 07ED C608 LDB $8 HOW MUCH TO MOVE
01087 07EF 308848 LEAX 72,X END
01088 07F2 3502 PULS A GET BACK CODE
01089 07F4 8147 CMPA $047 MOVE RIGHT?
01090 07F6 2724 BEQ MOVRGT YES
01091 07F8 2054 BRA MOVLFT NO; MOVE IT LEFT
01092
01093 07FA 3402 DINLIN PSHS A DELETE OR INSERT A LINE
01094 07FC 8D12 BSR LINSTR GET CURRENT LINE START
01095 07FE ECC030 LDD DSTART,U GET LAST LINE
01096 0801 C30730 ADDD $0730
01097 0804 1F01 TFR D,X STORE IT
01098 0806 C608 LDB $08 AMOUNT TO MOVE
01099 0808 3502 PULS A
01100 080A 8145 CMPA $045 INSERT A LINE
01101 080C 270E BEQ MOVRGT YES
01102 080E 203E BRA MOVLFT ELSE DELETE A LINE
01103
01104 0810 17FD0E LINSTR LBSR FINDPO GET CURRENT POSITON
01105 0813 E6C82E LDB CURCOL,U DELETE COLUMNS
01106 0816 50 NEGB
01107 0817 3085 LEAX B,X TO FIND LINE START
01108 0819 1F12 TFR X,Y PUT IT IN Y
01109 081B 39 RTS
01110
01111 *****
01112 * SHIFT RIGHT
01113 * B = NO. OF CHARS
01114 * Y = START POS
01115 * X = END POS
01116
01117 081C 10AFC032 MOVRGT STY WORK,U SAVE START
01118 0820 301F LEAX -1,X NEEDED FOR ALIGNMENT
01119 0822 3404 MOVR10 PSHS B SAVE COUNT
01120 0824 17FD73 LBSR MOVC10 MOVE CURSOR TO POS
01121 0827 17FC9D LBSR GETCHA GET A CHARACTER
01122 082A 3504 PULS B GET OFFSET
01123 082C 3414 PSHS B,X ADD TO POS
01124 082E 3A ABX
01125 082F 3402 PSHS A SAVE CHAR
01126 0831 17FD66 LBSR MOVC10 MOVE TO POS
01127 0834 3502 PULS A GET BACK AND
01128 0836 17FC6D LBSR PUTCHR MOVE TO NEW LOCATION
01129 0839 3514 PULS B,X GET BACK OLD POS
01130 083B 301F LEAX -1,X DECR POS
01131 083D ACC032 CMPX WORK,U DONE ?
01132 0840 2CE0 BGE MOVR10 NO
01133 0842 8E0000 MOVR00 LDX $0 CLEAR POSITION
01134 0845 3A ABX
01135 0846 1E12 EXG X,Y IN Y
01136 0848 17FC87 LBSR CLEAR GO
01137 084B 16FEC8 LBRA CTREND ALL DONE;CLEAR CONTROLS
01138
01139 *****
01140 * SHIFT LEFT
01141 * B = NO. OF CHARS
01142 * Y = START POS
01143 * X = END POS
01144
01145 084E AFC032 MOVLFT STX WORK,U THIS DOES JUST WHAT
01146 0851 1E12 EXG X,Y MOVRGT, ONLY IN REVERSE
01147 0853 3414 MOVL10 PSHS B,X
01148 0855 3A ABX
01149 0856 17FD41 LBSR MOVC10 GET A CHARACTER
01150 0859 17FC6B LBSR GETCHA
01151 085C 3514 PULS B,X
01152 085E 3404 PSHS B
01153 0860 3402 PSHS A
01154 0862 17FD35 LBSR MOVC10
01155 0865 3502 PULS A
01156 0867 17FC3C LBSR PUTCHR SAVE B CHARACTER BACK
01157 086A 3504 PULS B
01158 086C 3001 LEAX 1,X
01159 086E ACC032 CMPX WORK,U
01160 0871 25E0 BLO MOVL10
01161 0873 20CD BRA MOVR00
01162
01163 0875 2DD266 EMOB
01164 0878 WORDP1 EQU *
01165
```

```
00000 error(s)
00025 warning(s)
00078 02160 program bytes generated
00097 00151 data bytes allocated
0291A 10522 bytes used for symbols
```

Hard Disk Subsystem for SS-50 Computers

THIS PROVEN SUBSYSTEM ADDS HARD DISK SPEED AND STORAGE CAPACITY TO YOUR COMPUTER YET REQUIRES ONLY ONE SS-30 SLOT. SOFTWARE (WITH SOURCE) IS INCLUDED FOR YOUR CHOICE OF FLEX⁹ OR SK[®]DOS[®], OS-9[®] LEVEL I OR LEVEL II, OR OS-9 68K OPERATING SYSTEMS. THE SOFTWARE HONORS ALL OPERATING SYSTEM CONVENTIONS. THE SOFTWARE IS DESIGNED FOR THE XEBEX S1410 CONTROLLER INTERFACING TO ANY HARD DISK DRIVE THAT CONFORMS TO THE ST506 STANDARD. FOUR SUBSYSTEMS ARE AVAILABLE:

- 1) 27 MB (FORMATTED) CONTROL DATA CORPORATION WREN HARD DISK, XEBEX S1410A CONTROLLER, SS-30 INTERFACE CARD, ALL CABLES, AND SOFTWARE FOR \$2850;
- 2) 7.3 MB (FORMATTED) TANDON TM-603 HARD DISK, REST SAME AS ABOVE FOR \$895;
- 3) NO HARD DISK, REST SAME AS ABOVE FOR \$600; AND
- 4) SS-30 INTERFACE CARD AND SOFTWARE FOR \$200.

ALL PRICES INCLUDE SHIPPING. WE ACCEPT VISA AND MASTERCARD WITHOUT ADDING A SURCHARGE. TEXAS RESIDENTS MUST ADD SALES TAX. THE SUBSYSTEM MAY BE MOUNTED WITHIN YOUR COMPUTER CHASSIS OR IN A SEPARATE ENCLOSURE WITH POWER SUPPLY. PLEASE WRITE OR PHONE (INCLUDE YOUR DAY AND EVENING PHONE NUMBERS) FOR MORE INFORMATION. WE WILL RETURN NORTH AMERICA CALLS SO THAT ANY DETAILED ANSWERS WILL BE AT OUR EXPENSE.

WELLS WRITTEN[™]
ENTERPRISES

P.O. Box 9802 - 845
AUSTIN, TEXAS 78766

*** (512) 244-6530 ***



FLEX IS A TRADEMARK OF TECHNICAL SYSTEMS CONSULTANTS, INC.
SK[®]DOS IS A TRADEMARK OF STAR-KITS
OS-9 IS A TRADEMARK OF MICROWARE AND MOTOROLA