

the world of micros

Support for Motorola based computer systems and microcontrollers

Jason and the Internet...

Guess what guys, it is coming time to pay for Internet access!



Rick Ulland returns!
Starting anew with operating system nine!

BASIC programs for DECB users to type in! Display and edit MS-DOS and MS-Windows .PCX & .BMP graphics files on your CoCo3!

CONTENTS

<i>From the editor</i>	2
<i>From our readers</i>	3
<i>Jason and the Internet</i>	4
<i>Frank Swygert</i>	
<i>AT 306 Trials and Tribulations</i>	7
<i>Frank Swygert</i>	
<i>Operating System Nine</i>	8
<i>Rick Ulland</i>	
<i>Breaking the 64K Barrier</i>	11
<i>Allen Huffman</i>	
<i>The Zen of CoCo Programming</i>	16
<i>Chet Simpson</i>	
<i>Modify your RS-232 Pak</i>	17
<i>John Kowalski</i>	
<i>BMPREADR/PCXREADR</i>	19
<i>Stuart Wyss-Gallifent</i>	
<i>Advertisers Index</i>	21

Is your subscription up for renewal soon? Check the list on page 2 for your name!!!

!! LAST CALL !!

Don't forget... this may be the last chance (though I certainly hope not!) to go to a Chicago CoCoFest! It's just around the corner! Read the back cover for details!

POSTMASTER:

If undeliverable return to:
FARNA Systems PB
Box 321
WR, GA 31099

If your address is incorrect, send me a postcard!

the world of 68' micros

Publisher:

FARNA Systems PB
P.O. Box 321
Warner Robins, GA 31099-0321

Editor:

Francis (Frank) G. Swygert

Subscriptions:

US/Mexico: \$24 per year
Canada: \$30 per year
Overseas: \$50 per year (airmail)
Back and single issues are cover price. Canada add \$0.50 per issue (\$1.00 max) additional shipping, overseas add \$2.00 per issue. Bulk/newstand orders available.

Advertising Rates:

Contact publisher. We have scales to suit every type of business. Special rates for entrepreneurs and "cottage" businesses.

Contributions:

Any and all contributions welcome. Submission constitutes warranty on part of the author that the work is original unless otherwise specified. Publisher reserves the right to edit or reject material without explanation. Editing will be limited to corrections and fitting available space. Authors retain copyright. Submission gives publisher first publication rights and right to reprint in any form with credit given author.

General Information:

Current publication frequency is bi-monthly. Frequency and prices subject to change without notice. All opinions expressed herein are those of the individual authors, not necessarily of the publisher. No warranty as to the suitability or operation of any software or hardware modifications is given nor implied under any circumstances. Use of any information in this publication is entirely at the discretion and responsibility of the reader.

All trademarks/names property
of their respective owners

The publisher is available via e-mail at:
dsrtfox@delphi.com

ENTIRE CONTENTS COPYRIGHT
1996, FARNA Systems
(authors retain copyright to articles)

A message from the editor...

Let's see... there is a lot going on in the world today! I almost got sent to Kuwait. A camp maintenance team will have left from here to a remote camp for 90 days by the time you read this. At first I definitely wasn't going to be on the 35 man team, then they decided I was, then the powers that make all decisions decided I wasn't going after all! Seems we had a volunteer to go, he had to back out, I was next on the "non-volunteer" list, but then another fellow decided he really wanted to go. Things happen! I'm just glad the decision is cast in stone now.

As long as I'm in the military, something like this could happen. And I'm more or less prepared for it. In the case of a 90 day excursion like the one mentioned, readers may have a long period between issues. You wouldn't miss an issue, it would be made up later. In the case of a longer excursion, I now have a laptop on which I can prepare most of the material. My wife and maybe a colleague or two would make up the rest, and my wife would handle printing and mailing. So some dates might get pushed or shuffled around, but you will still get your magazine!

I'll be in Chicago once again this year! I hope to see many of you there also. I should, or at least expect to, have an AT306 up and running. I haven't had much time to sit and play with one yet, as we have run into a few problems. As you can see in the article in this issue, I believe the problems have been corrected and we are well on our way to a stable platform.

There was one little problem with this year's Chicago CoCofest... the Holiday Inn had to give up most of our rooms!!! What happened was that the Holiday Inn franchise folks issued many of the larger holders an ultimatum: refurbish rooms now or loose your franchise. This came on unexpectedly, and caught the Elgin Holiday Inn by surprise. They had to drop our rooms because or the time-frame in which construction was mandated to begin. They had little choice in the matter... their franchise was due to be renewed, so they had to act fast! Unfortunately, ours was the only block of rooms available. Others were affected also, we aren't getting picked on! The hotel officials didn't know this until early this year... to late to change fest lo-

cations or Glenside would have done so. Accomodations were slim in the area due to a toy convention, but the Red Roof Inn was able to help out. The Red Roof is about 10 miles away, but Glenside members and vendors with extra room will be helping those who need a ride between the Holiday Inn and Red Roof, especially those who fly in. Details for making reservations are on the back cover.

I'm going to continue printing the names of those who recently renewed their subscriptions. I'm also going to print the names of those who have subscriptions about to expire. It seems that many miss the handwritten note on the mailing label. Note that this is just a friendly reminder, I am not trying to embarrass anyone! Some time periods will have a larger number of renewals due than others (such as early spring.. as this list shows!), so don't be alarmed if one issue has a long list and another a short one. Right now, I am averaging 75-80% renewals. Of the 38 listed below, I expect at least 30 to renew. We'll all see in the next issue! And remember, by renewing, you do your small part in keeping the CoCo community alive and well!

I would like to thank the following people for renewing their subscriptions:

Henry Harwell	Kurt Ryman
Don Guido	James Ogden
Allen Morgan	Pete Bumgardner
Alan Gordon	W.R. Wildman
Robert Sullivan	Mark Stainer

The following have renewals now due or to be due after the next issue:

Edmund Bassick	Richard Batt
George Bethea	Carl Boll
David Breeding	Larry Bryant
Hugo Bueno	Earl Casper
Malcolm Cleveland	J.W. Cross
Ralph Ferringer	John P. Francis
Robert Gibon	Brian Goers
Scott Griepentrog	Dale Hawley
Mike Holick	Allen Huffman
Brother Jeremy	James Jones
Keith Kounovsky	Ron McCauley
Ron Mehn	Barry Miller
Don Peters	Jane Elliot
Rick Porter	Jason Reichard
R.E. Rutherford	John Schaefer
Erik Seielstad	Daniel S...
Wayne Thompson	Stefan Tomaski
Earl M. White	Donald Yehli
Paul Zibalia	Ken Scales

Messages from our readers...

Microware... you listening?

First off, I'm writing this letter so you will have my new address. As one of the original subscribers, I don't want anything to hinder me in receiving my next issue of "the world of '68 micros".

While I'm writing to you, I have a question on your Jan/Feb '97 AT306 article. If I put 64MB (4x16MB) in the motherboard as you stated was possible, is that feasible? I thought that the 68306 could only address 16MB of memory due to its 68000 core. Or are you telling us that just in case Carl Kreider expands this board with a more powerful MPU that can map the 64MB memory? Don't take me wrong, I'm trying to save up the \$500 so I can finally finish off building one. I'm down to buying the motherboard, video card, and a 17" monitor. OK, the 17" monitor is going to cost another \$700, but I think it's worth it. I have an old Wyse terminal I can use until I come up with the extra dough.

1) As a multi-OS user both at home and at work, if there is news or ideas from other platforms that can be remotely tied to 6809/68xxx or OS-9/OS-9000 I say print it. I too saw information on the JAVA version of Corel Office being ported to OS-9. When I went to Corel's web sit to pull the pre-beta version, it would only work on a Mac or PC (maybe a Sun box) and only if you had a JAVA browser. Well, my OSK boxes at work, the ones that run and monitor surface mount chip machines, don't have a JAVA interpreter let alone a web browser. But trying JAVA Office on a PC was an experience! This is what JAVA was written for, not for cheap animation on web pages. One code running under different platforms without recompilation.

2) I'm afraid that the Atari and Amiga crowds won't show much support. The Amiga crowd is still waiting on the next generation of Amigas, and the Atari crowd has an 8-bit/ST split.

3) I think Microware should create a small personal hobbyist division or have a closely tied company to Microware to sell and support the hobbyist/programmer. If I could go to a PC catalog and buy OS-9000 ready to run on a typical PC or a Mac catalog for a PowerPC version, then these machines would have a real option for OS-9 instead of NT or the Mac OS. Those who want or need more won't waste their money on SCO UNIX. The more people that are familiar with OS-9 as a development platform than that of a solution for embedded consumer products, the less hacked up "real-time" UNIXes that run on i386 are out there trying to handle what OS-9 was designed for.

4) A consumer ready OS-9 bundle should include all the drivers a typical PC or Mac has... sound, ethernet, video for common cards and/or a driver convertor (program that reads the DOS driver, figures out what is DOS and

what is being passed to the device/controller, and change it into an OS-9 module and driver). If I could buy an OS-9 system with video, sound, and ready to run software, I would buy a few for development and emergency replacement. That way I could develop on a non-live machine and also know that within a few hours we will be limping but not shut down.

David Hazelton
672 Auburn Street
Manchester, NH 03103

That was quite a letter, David! Nice to hear from you. The AT306 should be able to address 64MB with no problems. See the AT306 article in this issue.

1) Now all we need is an inexpensive JAVA interpreter and/or browser to run real applications! Unfortunately, I doubt we will see them in anything but future releases of OS-9000.

2) You are correct about the Atari and Amiga people. While the next generation Amigas and some clones are out now, they are not interested in any support issuing from the US (a German company bought Amiga, and most Amiga support now comes from Europe). I have had some contacts from both sides of the Atari camp (8 bit and ST), but none have been able to generate enough interest to be worth pursuing.

3) I think the best we can expect from Microware is that they support a couple of hobby businesses, such as Carl Kreider's operation. He was able to get a very attractive licensing deal, better than most industrial users would be able to manage, simply because Microware does wish to encourage hobbyists to some small degree. We are close to providing a turn-key system... see the article!

4) David, if you're a good enough programmer to write that driver convertor, my hat is off to you! That would take quite a bit of logic and near artificial intelligence to pull off!

Overall, I think your ideas hit the spot. Hopefully, Microware is listening and Carl, Bill Wittman, and myself can fill some of the gaps.

Will you be here next year?

Just received the Nov/Dec 1996 issue of two 68' m and the mailing label states it is my last issue. This is a inquiry to see if you still plan to publish the magazine for at least another year. If yes, then my re-up check will be in the mail as soon as I receive your reply.

I have enjoyed two 68' m for several years now and hope you are intend on continuing. There are a few hard core members left in our local CoCo club (Broome County CoCo Club). We all have other machines (MSDos or in my case OS/2) but still use our COCOs. Unfortunately, we are all end users so no articles will be coming from any of us.

Keep up the good work and I hope to hear from you soon.

Don W. Guido
dguido@binghamton.edu

I'll definitely be printing for another year or two! The only thing being discontinued is the "microdisk". There are only a few people who order it, and it just isn't worth the time. I will still honor requests for specific programs or files that someone is having a hard time finding and that are mentioned in the magazine. There will be a \$5 charge per disk, with a limit of three programs/files per disk.

Glad to have you back on board! And remember, even users can write articles. How do you use your machines? Any tips or tricks with certain programs you want to pass along? There is a LOT you guys could write about!!

CoCo is more fun...

No one seems to understand why people stick with the CoCo. My first computer was a CoCo, and I couldn't have been happier! I had a CoCo2 w/ Extended Basic, 2 5 1/4 floppy drives, a multipak interface, a sound card, a DMP110 printer, and a 300 baud modem. Unfortunately, I don't have the computer anymore. I ran into financial problems and had to sell in the late 80's. I would love to have the old machine back.

Sure, now I run a 486 DX2/80 with all the bells and whistles and a color printer. (Although the system was out-dated practically the day after I bought it) It does everything I want it to do. BUT, With this computer, I simply run programs, and do a lot of system tweaking. But there has never been a computer that I had more FUN with than the CoCo. (I used to write programs for that machine all the time. If I couldn't buy a program to do something, I'd write a program to do it!)

Reading the WEB pages and this list, I realize there are so many things out for the CoCo that I never imagined possible (a hard drive on the CoCo!?) I am very pleased to find more people that enjoy the CoCo enough to stick with it.

Don

There are still a few "CoConuts" about, Don! Most of us will agree, the CoCo is much more FUN than any MS-DOS/Windows based PC or Mac ever has been!

continued on page 18

Jason and the Internet

Frank Swygart

Most of us know we'll have to eventually pay for the 'net... and maybe soon!

As much as I need them, I have probably lost a subscriber. All because I didn't agree with their point of view. In fact, I'd say I more than disagreed... I took their point of view (and that of some of this person's friends) as almost ludicrous. I guess it wouldn't have been so bad had not they thought I'd be in support of them... I don't think the thought that I may disagree ever crossed this person's mind. The point was that telephone companies shouldn't have the right to charge for Internet access. The reasons I disagree are clear and logical. I'll explain them later, of course, but some background information is the first order of business.

Back in the early eighties, some telephone companies wanted to make additional charges for data access. This was when home operated bulletin board systems (BBS's) were gaining in popularity. The telco's ideas were that data communications required cleaner lines,

and they would eventually have to upgrade lines to provide ever increasing access speed. But the telco's were defeated in this proposition.

The government got quite a bit of mail from the few of us who actually ran and used this system. Our arguments were that computer communications accounted for a very small percent of traffic over the telephone lines and that the telco's would have to upgrade their lines eventually anyway... lines that were not running anywhere near capacity in most areas. Not only that, but the limitations of data transfer was in the hardware, it would be quite a few years before the capabilities of the existing infrastructure would be exceeded or taxed in any way by telecommunications equipment. Telephone calls were already charged at per minute rates, that alone would limit the amount of time a person stayed on long-distance lines (remember, at this time BBS's were few and far between, not many of us were lucky enough to have a large local BBS!). To top this argument off, the government was interested in fostering digital communications. As long as there was a populace, even a small percent, interested, companies would devote time and research to developing faster, better ways to transmit information. So we, the BBS users and pioneer electronic communicators won this great battle.

Many people compare the battle of the 1980's with the current debate over whether telco's should charge access fees to the Internet. But there is a totally different set of circumstances today than it was back then. The digital communications industry is no longer a fledgling. We aren't talking about a few hundred thousand hobbyists wanting access anymore... we are talking about MILLIONS of people. And we aren't talking about an infrastructure with excess capacity... we are talking about a system that needs upgrading YESTERDAY, not tomorrow!

The phenomenal growth of the Internet has caused all this change. For the average person, the telco's proposed charges would mean an additional \$10-\$12 per month on top of the \$20-\$25 currently paid for unlimited access. \$30-\$40 per month is hardly something to argue about, not when there is a real need. How many of you have jumped on the Internet during busy times only to have to wait minutes for a single page to come up? How many of you are in areas where 9600 or 14.4K baud is the fastest you can think about getting on? How many of you can't get on the Internet through a local number at all? These are the things the telco's want this extra money for. They don't want it just to gild their pockets. Telephone companies are utilities, and as such are governed by federal regulations as to how much profit they can make. The bottom line is that the telco's can't make any more money, and they have to show justifiable reasons to raise basic rates or create additional charges. They have justifiable reasons this time.

There are millions of Internet users right now. Use is expected to double again in the next year or two. Part of the reason is the relatively new "Internet TV" boxes now on the market. This puts the Internet into the reach of many non-technical people. Sales of these are expected to really take off. Those people scared of computers but curious about all the Internet hype will get them. After all, they are no scarier than a TV remote or a video game pad. Sales of computers have soared over the last year or two also. Both the Internet AND Windows95 have caused this. I do not like Win95 a whole lot, it is far from the best operating environment (it is much more than a system!) available. The installed user base, however, is extremely large. This makes it easy for a beginner to get help. And Win95 is much better and easier for the first time computer user to learn than any sys-

From: Dennis Bathory-Kitsz

Hi folks! I've been hiding out in Vermont, but since it's the 10th anniversary of my company Green Mountain Micro's demise, I thought it might be time to put in an appearance here.

About 150 copies of 'Learning the 6809' (book only) remain, which I'd be happy to offer at \$10 postpaid to anyone interested. If at least 10 people also want the original tapes, I'd be pleased to make up a set of those as well.

One of these days I'll tell my own tale ... amusing indeed...

Dennis Bathory-Kitsz
RD 2 Box 2770
Cox Brook Road
Northfield, Vermont 05663

<bathory@maltedmedia.com>
Malted/Media:
<http://www.maltedmedia.com/>

tem before it, save the MacIntosh OS. I have used both, and I think it is fair to say that Win95 is more "Mac-like" than any other system out there. And that is no coincidence.

The young man I mentioned earlier sent me an e-mail message speaking out against additional Internet access charges. I thought this absurd. Absurd that anyone should be raising such a fuss over it. And I told him so. He sent me some replies from a friend of his, and I blasted him also. There are actually people out there who think everything should be free! I pointed out that this was a totally different situation than that of the 80's. The Internet and the US telephone system is heavily burdened by the sudden, unexpected surge of digital traffic. Indeed, low-cost "unlimited" access is part of the problem. Back in "the old days" (just 2-3 years ago now!) we had to pay for every minute of access time on most systems. So we wisely apportioned our time spent. Not any more. The computer and Internet have taken on the role of an entertainment medium, not a research or work medium. Check out some of the "chat house" lines and one will find all kinds of low-lifes hanging out with nothing better to do than hang around and insult the intelligence of most people. Seriously, try getting on "www.chathouse.com", for one.

Not everyone just "hangs out" on the Internet. There is a lot of good sharing of information. Some of the chat lines are a great way to meet other people interested in particular subjects. There are many isolated communities that the Internet opens up. And it should. The Internet should open minds and help them to grow. And it does just that for many of us. But as long as there is an audience for what most people consider "bad" or "evil" things, then there will be some exploitation of the fact that some people want it. And I have to defend the lack of censorship... after all, I am a US military member, sworn to uphold the Constitution of the United States of America. That document states that I have the right to print what

I am printing now, and for others to print, say, and broadcast their own opinions, regardless of what I think about it.

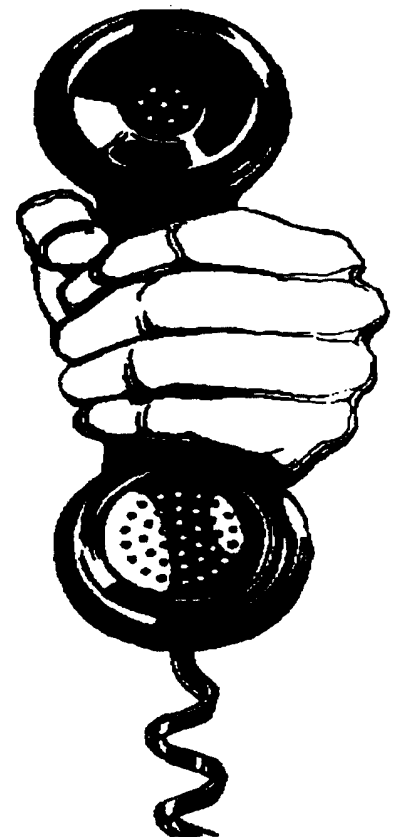
The Internet is no more the "root of all evil" than broadcast TV or printed media. Parents who often don't want to be parents lament that they are afraid of what little Johnny or Betty will find on "the big bad Internet". Guess what folks! Kids will be kids! They need adult supervision! You HAVE to say NO sometimes, and you HAVE to keep an eye on them! How many of you remember sneaking copies of Playboy off and oggling and reading? How many of you remember sneaking off to a friends house to watch dirty movies, or into the X-rated drive ins? I remember some of this, and I'm sure many of you do too.

Experts are unimpressed with the reports of additional telco charges for Internet access. So are those who follow the computer communications industry and its growth. We (yes, I consider myself one of those interested parties) have not only been expecting something like this, but are actually glad to see it come. This means that lines will be upgraded faster and access will be more universal across the nation. The telco's work on five and ten year expansion plans. The sudden demand created by the Internet phenomoena has wreaked havoc with most of those plans. The long range plans had to be revamped comparatively overnight by most telco's. There is no way they could have accurately predicted the explosive growth of the Internet. Consequently, there is no way they can adequately accomodate the current, much less the future, demand without greatly accelerating their expansion plans. And expansion costs money.

For the past two years, industry analysts have been wondering when it was going to get down to costs. Who will pay for the Internet's growth? Who or what will pay for the Internet? well, the time is coming, and soon, when it all has to be paid for. We have used up all the excess capacity, and now real expansion is necessary to keep the In-

ternet from collapsing. There will be a few who will drop the Internet because of the higher costs... for a short while. The most "damage" that I believe will occur is that people will once again have to consider what they are doing and why. They will have to justify some of their on-line time and expenses. Is being responsible while using something that bad? Where are we all going? Is Joe and Jane America THAT spoiled?

*For more reading, try the February issue of "Computer Shopper", page 594, "Bandwidth Blues". For those with Internet access, this article is available at: www5.zdnet.com/cshopper/content/9702/cshp0038.html.



The bottom line: With more users every day, telephone companies will HAVE to install newer, faster equipment to keep up with demand.



FARNA Systems

Your most complete source for Color Computer and OS-9 information!

Post Office Box 321
Warner Robins, GA 31099
Phone: 912-328-7859
E-mail: dsrtfox@delphi.com

ADD \$3 S&H, \$4 CANADA, \$10 OVERSEAS

BOOKS:

Mastering OS-9 - \$30.00

Completely steps one through learning all aspects of OS-9 on the Color Computer. Easy to follow instructions and tutorials. With a disk full of added utilities and software!

Tandy's Little Wonder - \$25.00

History, tech info, hacks, schematics, repairs... almost EVERYTHING available for the Color Computer! A MUST HAVE for ALL CoCo aficionadoes, both new and old!!!

Quick Reference Guides

Handy little books contain the most referenced info in easy to find format. Size makes them unobtrusive on your desk. Command syntax, error codes, system call, etc.

CoCo OS-9 Level II : \$5.00

OS-9/68000 : \$7.00

Complete Diato Schematic set: \$15

Complete set of all Diato product schematics. Great to have... needed for repairs!

"A Full Turn of the Screw": \$20

Lots of CoCo info, projects, and tutorials.. by Tony DiStefano

"Inside Diato's 2 Meg Kit" : \$10

Schematics and explanation of how the 2 meg CoCo 3 upgrade works.

SOFTWARE:

CoCo Family Recorder: Best genealogy record keeper EVER for the CoCo! Requires CoCo3, two drives (40 track for OS-9) and 80 cols. DECB: \$15.00 OS-9: \$20.00

DigiTech Pro: \$10.00

Add sounds to your BASIC and M/L programs! Very easy to use. Requires user to make a simple cable for sound input through a joystick port. Requires CoCo3, DECB, 512K.

ADOS: Most respected enhancement for DECB! Double sided drives, 40/80 tracks, fast format, many extra and enhanced commands! Original (CoCo 1/2/3) : \$10.00

ADOS 3 (CoCo 3 only) : \$20.00

Extended ADOS 3 (CoCo 3 only, requires ADOS 3, support for 512K-2MB, RAM drives, 40/80 track drives mixed) : \$30.00

ADOS 3/EADOS 3 Combo: \$40.00

Pixel Blaster - \$12.00

High speed graphics tools for CoCo 3 OS-9 Level II. Easily speed up performance of your graphics programs! Designed especially for game programmers!

Patch OS-9 - \$7.00

Latest versions of all popular utility and new commands with complete documentation. Auto-installer requires 2 40T DS drives (one may be larger).

NEW ITEMS!!!

FARNA Systems is pleased to announce that we are now distributors of the following, formerly from Northern Exposure! Note: If you never received your order from NX, send a copy of your cancelled check along with \$5 to cover S&H and I'll fill the order!

Nitro OS-9 : \$35.00

A complete rewrite of OS-9 Level II that takes advantage of all features of Hitachi 6309 processor. Easy install script! 6309 required.

TuneUp : \$20.00

If you don't have a 6309, you can still take advantage of some of the Nitro software technology! Many OS-9 Level II modules rewritten for improved speed with the stock 6809!

Thunder OS-9

Shanghai OS-9 : \$25.00 each

Transfer your ROM Pack game code to an OS-9 disk! Please send manual or ROM Pack to verify ownership of original.

Rusty : \$20.00

Launch DECB programs from OS-9! Allows loading of some programs from hard drive!

FARNA Systems AT306 Based Computers

Complete computer systems based on the AT306 board from Kreider Electronics. Systems are completely setup and ready to go. Just add a VGA monitor (or we can supply that too)!

Both systems include:

16 bit PC/AT I/O bus with five slots
MC68306 CPU at 16.67MHz
4 30 pin SIMM sockets
IDE Hard Drive Interface
1.4MB Floppy Drive
Two 16 byte fast serial ports (up to 115K baud)
Bi-directional parallel printer port
Real-time clock
PC/AT keyboard
Desktop Case and Power Supply
(mini-tower case optional, no cost!)

BASIC (resembles Microsoft BASIC)
MGR Graphical Windowing Environment
with full documentation

"Personal" OS-9/68000 V1 3.0

(Industrial with RBF)

Drivers for Tseng W32i

and Trident 8900 VGA cards

Drivers for Future Domain 1680

and Adaptec AAH150x SCSI cards

FARNA-11123 Includes:

2MB RAM
300MB Hard Drive (was 200!)*
Trident 8900 1MB Video Card
\$960.75

**This is the SMALLEST amount of formatted space available.*

Prices fluctuate - we get you the largest drive possible for the money allotted!

FARNA-11225 Includes:

2MB RAM
500MB Hard Drive*
Tseng W32i 1MB Video Card
\$1114.47

HACKERS MINI KIT (FARNA-11100): Includes AT306 board, OS-9 and drivers, utility software, assembly instructions/tips, T8900 1MB video card. Add your own case, keyboard, drives, and monitor! **ONLY \$500!**

Call for a quote on different configurations and components.
Warranty is 90 days for labor & setup, components limited to manufacturers warranty.

Microware Programmers Package -

Licensed copies of Microware C compiler, Assembler, Debugger, Basic, and many other tools!

With system purchase: \$65.00 Without system: \$85.00

Many other utilities and tools

AT306 Trials and Tribulations

Frank Swygart

Murphy's law presides over our new OS-9 computer system!!

That darned old Murphy gets us every time, doesn't he? Right when you think everything is fine and you're ready to go on to another project, BAMM! he hits you!

Carl Kreider asked me if I would be interested in selling AT306 boards and/or systems before he started building them. We had quite a bit of communications then, with some of my suggestions having an influence on the finished product. I initially declined selling the boards myself, preferring to advertise for the three or four other dealers (BlackHawk, Bill Wittman, and a fellow whose name I can't recall right now in California.... and BlackHawk intended to sign on more dealers). I did manage to get put on the mailing list dealing with programming and development issues concerning the AT306.

There were at least two developments that made me change my mind about selling systems. The first was that after following the mail list for about a year, it seemed that most of the bugs had been worked out of the system. Everything seemed to be up and running just fine. The second was that BlackHawk was unable to get enough funding to create the small dealer network planned.

With only a couple of friendly competitors spread over the map, there seemed to be enough of a gap in the market for me to enter. And it seemed that the experts had already worked everything out. I should be able to purchase components and easily assemble a running system. I did say "should".

My strategy was simple: put together a base system that was ready to go straight out of the box. This system would be warranted to run "as set up". By this, I mean as it was when you turn it on. This would be at least three text video windows and one graphics window running the MGR graphical user interface. The user would get the system with everything on the hard drive ready to roll, the original program disks AND a set of backup disks that would restore the system to an as shipped condition. I'm not a programmer, and am a fledgling OS-9 user myself, so I had to wait until the system appeared to be stable enough for a novice to use and for me to setup and offer some support. The time seemed to be about right...

Shortly after advertising systems for

sale, I received a couple of orders. One was for a complete system ready to roll, the other for a board and video card. Both were paid in full at the time they were ordered. I immediately called Carl and confirmed availability of a couple boards then ordered them. At the same time, I called a mail-order PC parts supplier and ordered several Trident T8900 video cards and the components to build a complete system.

In about three weeks, everything was at my house and ready to be assembled. The board and video card were sent to their new owner posthaste. That next weekend, I started assembling the AT306.

In the last issue, I described the hardware assembly and setup. Piece of cake! I'm currently serving as a PC technician for the USAF, so I've seen lots of hardware over the past two or three years. Anyone who has ever swapped motherboards in a PC or stripped their CoCo down can easily put an AT306 system together using common parts. That part of the design criteria was met!

Booting the system was another story. Upon turning the system on, all I got was pink and blue vertical lines. Not a good sign. So I inserted the boot disk. No change. Next step: double check all connections and settings. Everything checked out okay.. time to call Carl.

Carl is equally perplexed with the problem. It seems that the video card is not initializing. It seems that the video chipset might be the problem. These video cards have only two integrated circuits (chips) on them. The ROM code is built into the main video chip.. the chipset reads T8900D-R. Carl has had a problem with a developer's T8900D video card also... he'll have to do some experimenting and make some changes to the video driver.

After two weeks head scratching, hair pulling work, Carl finally has a video driver that works with all Trident T8900 B, C, and D cards. He makes up a bootable disk for me and sends it. A few days later, I pop the disk in and give it a try. Guess what? Pink and blue vertical lines. I try a genuine Trident manufactured T8900D card with a Trident 4.3 ROM BIOS. This one does no different than the others.

So I call again. This IS a perplexing problem! I have an extra video card, so I send it to Carl. Another two weeks of hair

pulling (Carl is about bald by now... I think I'll see what a hairpiece would cost...) and Carl has a new boot that works with everything! It seems that Trident accesses a register or two in the EGA portion of a PC's BIOS ROM. Since the AT306 isn't a PC, it doesn't have anything in these memory locations, or at least not what the Trident ROM expects to find.

This past week, I was at my father's home in South Carolina helping him put together a decent system for my nephew to use in college. Between our respective junk boxes, we managed to put one together. But the real find was a Trident T8900B card at my father's. And it didn't have a Trident ROM!!

Well, I thought that my problem was fixed for the moment. I've had this fellow's machine for about three months now. I keep sending him e-mail appraising him of every snag and development, and he has been very patient and pleased that I have kept him up-to-date, but I'm sure his patience is wearing thin! Even if he isn't, I'm starting to feel awful about keeping his machine so long. It doesn't look good for me, and I wouldn't be surprised if he were beginning to wonder if I ever intend to send him a computer (this isn't the case, of course, nor does he think this, but...!)

I tried the T8900B card just today, a few hours before I wrote this (08 March). I still get pink and blue lines, but not in exact vertical bars this time. I assume this is in part due to the different ROM, part for the fact this particular card only has 512K on it instead of 1MB as the drivers were written for.

I just got off the phone with Carl. He has discovered that the last few boards that left the shop had a bad PAL chip in them, and BOTH of the boards I purchased have the bad chip!

A PAL is a Programmable Array Logic chip. It can be setup to mimic the functions of a whole handful of small chips, transistors, diodes, etc. Makes the physical act of building circuit boards much easier. But if any of you have ever "burned" EPROM chips for your CoCo, you will know that it is possible to make a minor mistake, get a bad EPROM, or for any number of things to go wrong.

continued on page 18

operating system nine

Rick Ulland

The engine

Editor: I have to take a little space to welcome Rick back! He was moving, changing "regular" jobs, and everything else for a while and needed a break. Alan Dekok had just sent me a few articles on Nitro, so that fit the bill just right! Rick hadn't intended on staying away for three issues, that was my decision, since I wanted to run Dekok's articles in this column. Glad to have you back Rick! Hope to see you in Chicago this year! Which reminds me, time to make reservations!!!

Over the last few months, an interesting thing has happened in CoCo land... we are beginning to pick up new users. I'm not sure if this means the machine is now old enough to be sexy, or if the world is just coming to it's senses, but there you are. There are even a few adapting Tandy Level 1 for use on various home-brew systems! I want to do my part to encourage the new folks, and since we are starting a new series of 'os-9' this is a good place to cover the whys, wheras, and hows of obtaining and setting up a full system.

In this day of Pentium powered wonderboxes why would anyone want a CoCo? Well, with OS-9 it occupies two niches the PC isn't particularly good at. First, it's a little bitty (cheap) minicomputer. There are limits- the 6x09 isn't particularly fast or very large in terms of RAM access, enjoying the 64K limit common to 8-bit cpus. But limits are made to be exceeded and even though individual programs are still confined to the 6x09 framework there can be dozens of frames (even the opsys gets a separate 64K) and in the CoCo3, each of these gets another 64K for graphics. With OS-9's filesharing and security you can get multiple users on the same data set by adding nothing more exotic than a few dumb terminals and a little more hard drive.

Paradoxically, it's other strength lies at the opposite end of the scale, in single board controllers limited to a few kilobytes of ROM. Plugging homemade interfaces into a \$2000 PC can be a harrowing experience! In contrast, a \$10 CoCo2 comes with a much more capable operating system and can almost be considered disposable. In fact, this is why I bought my first CoCo, as an experimental robot controller. OS-9 would allow a simple scheme of multiple 'twitch drivers' for the body parts, with supervisor code overlaid as a separate program. As an added bonus the expanded CoCo3 as development system means you don't have to choose between cross assembling or trying to develop software in a microscopic environment.

If you need to do either of these things on a budget, a detailed list of OS-9's software features, coupled with a price list, makes the CoCo sound like a good choice. Then reality bites-

the CoCo itself isn't a very robust computer. It was engineered a little on the cost-effective side and aimed at the 'introducing computers' market. It can be rigged as a serious OS-9 platform, but it has quirks which should be allowed for. This column describes what a practical CoCo system consists of.

Where to get it:

CoCo parts are getting a little hard to find now. To scan the net, join the CoColist (send a message to listserv@pucc.princeton.edu with the word 'subscribe coco your name' as main body of email. Send a followup letter with 'help'. The humans this listserv serves are at coco@pucc.princeton.edu). Used *ware lists pretty often, at very reasonable prices. There are also a few newsgroups worth checking out- if you don't want to join the cocolist right away, it's echoed to bit.listserv.coco (post direct to humans at coco@...) and there is a general Tandy newsgroup in comp.sys.tandy. Don't overlook the local thrift shops- recently CoCo3s have been appearing alongside the perennial CoCo2s. Another major item you can pick up locally is a (very) used XT or AT system. For \$20 or \$30, you'll get a switching power supply, XT keyboard, and maybe a small MFM hard drive. For the adventurous, a case for re-packaging into as well. As you'll see, these are prime OS-9 goods.

The first few hacks:

The system as a whole is going to be large- probably too large to comfortably inhabit a desktop. It's a bit messy as well, with carts and cables towering proudly in front of the monitor's screen. While it's possible to 're-pack' the whole thing into the XT case, that isn't the only option. Avoid the major headaches of a repack by simply moving the mass off the desktop! All you really need to see is the monitor, keyboard, and floppy drive.

The monitor and floppy already have long-ish cables- the addition of a simple keyboard extender cable will let you stash the actual system away in a safe place. For years, I had a stock CoCo and Y-cable rig (complete with hard drive) screwed to a piece of plywood, hanging on the back side of my desk.

While a keyboard cable will clear off your desk, it still leaves a piece of 'toy computer' hardware in plain sight- the keyboard, resplendent in your home made case. Worse, the CoCo keyboard is based on a teletype, not the typewriter most people are accustomed to. If you have to do any typing besides on the CoCo the difference can result in lots of typos, or at least a reduction in speed. What you'll want is a Puppo keyboard adapter. These used to cost about \$80 new, but by now you'll have to convince a longtime CoCo user to part with his.

They used 'XT' keyboards so the keyboard from the parts XT will work fine. It's difficult to find a brand new keyboard to use with the Puppo, since most hi-end and all the bargain units available nowadays are AT only. For most Puppo's, you'll want a switchable unit with a physical switch, or hunt the surplus mail-order places for an original 86 key unit.

If a large system is planned, with hard drives or more than two buss cards, the CoCo's power supply(s) will be too small. It's best to add a PC style switching power supply, like the one from your parts box XT. In addition to the extra current capacity, the switching supply is less affected by it's input voltage. The optimum is a rewiring of the CoCo and multipak to use the new power, which can give a very noticeable increase in reliability. Even if you are not up to rewiring the main boards, using it to power all the disk drives will be cheaper and simpler than dedicated external floppy and hard drive cases.

White Plastic Boxes

Starting with the CoCo itself. When the old warhorse dies, you can't just take the backup tape to a buddies house or pop out and pick up a replacement motherboard. The only way to ensure timely access to your data is a second CoCo! Find one before it becomes a problem. Usually a used unit will turn up locally, and at the local garage sale you might not have a monitor handy. You can check for basic operation by plugging the unit in and typing MOTORON. Listen for a second, internal click from the cassette relay as you press the enter key. If a questionable CoCo responds to MOTORON (and the related MOTOROFF) with relay clicks, you can reasonably expect it to be a working unit.

Given the chance, next deal with Tandy's former QC problem. All CoCo's are not created equally and some units are much more desirable than others. The first test is for GIME version- there were several problems with the 1986 edition of this chip, the upshot being the oldest CoCo3s have irritating 'features' and can't run some software. With the top off, you can just check the copyright date on the GIME (a large, square chip to right front of the mb) but you don't always need to dig so deep. If you can see the unit running a game, look for 'sparklies' (noise that looks something like a worn video tape) when the image on screen changes radically. No sparklies, no problem.

Even with an 87 GIME, there are other pitfalls so you'll want to determine the overall quality of the machine. The best overall quality indicator is heat- while all CoCo's run a little warm, some of them carry this to extremes. Let the machine run five minutes, then check the cooling slots that run along the top.

Some warmth on the left side is expected, but 512K units with substantial heat right in the middle (by the badge) are good machines to pass on.

The only other variable in a stock CoCo3 is RAM size. You'll need 512K (or more) for serious OS-9. The original 512K upgrades were made by dozens of vendors, and used ones are still available (CoNect is one source), but cheaper as part of a used machine. Look through the cooling slots for an add-on board just behind/ beneath the 'Tandy' logo- it will rise above the motherboard, about the level of the top of the metal TV can that lies just to its right. The Tandy RAM upgrade (you'll see a foil shield, instead of rows of memory chips) is probably the least desirable since it's upside down layout runs even hotter than average. It's usually OK, but hot climates or hardware hacks that add to or speed up the motherboard can push one into unreliability. Letting the CoCo hang vertically or sit open in an out of the way place might avoid unpleasant-ries but if you have a choice of equipment, keep this potential problem in mind.

In addition to the 512K units, DISTO made two larger RAM upgrades of 1 and 2 megabyte. The two meg version is preferred both for it's size and the relative ease of transplantation when the original CoCo dies. They are identifiable by an add-on board over the cpu and either a second card of DIP RAM (1meg) or a single memory card with SIMMs (2meg). Rights to the 2meg version have recently been acquired by Monk-O-Ware and new ones will probably be available once more in 97.

There is one other significant upgrade to the motherboard itself- the 6809 cpu can be replaced with a newer version, the 6309. This chip normally emulates a 6809 exactly, but with matching opeys upgrades (Nitro, available from FARNA Systems) it comes into it's own- the effective clock speed can be doubled. This isn't an upgrade for the casual tinkerer, since you have to de-solder and socket the cpu, a 40pin dip. If you manage to find a used CoCo with socketed cpu, grab it! One last note: the 6309 has a failure mode even longtime CoCo tinkers might not have seen- an apparent failure of the floppy controller. Anytime a 6309 system displays doubled characters in a DECB directory listing, replace the cpu. My theory is that like most CMOS devices, it uses built in diodes to protect it from static. The halt line (which has NO buffering) is then disabled when it's diode gets popped, leaving no other symptoms since the cpu itself isn't damaged.

Moving on, the next requirement is a buss. The built in port has shortcomings beyond the fact there is only one slot. First, it's practically unbuffered and very easy to overload. Except for the CoCo1, it doesn't fully implement the CoCo standard because the bipolar 12 volt supply used by some add-ons (serial ports, some hard drive controllers) isn't present on the motherboard. Still, the easiest, cheapest

way to get more buss slots is a Y-cable, and Small Gfx sells the bits. How big? Try to keep the whole thing very short- 6 inches or so. Common wisdom says two slots, but some have worked three successfully. This isn't always a simple yes/no proposition- instead, to much Y-cable might only cause the modem to drop a character now and then, or a disk to occasionally go bad for no apparent reason. There is no simple maximum spec due to the variability of the CoCo- some machines exceed the minimum specs more than others and the marginal systems are easily pushed over the edge. Cool running DRAM usually means the unit will take more strain! Across the board, hardware of later manufacture is more tolerant of Y-cables. Not only are late model motherboards more consistent, but late model add-ons often incorporate additional gating or timing fudges.

The best buss solution is Tandy's MultiPak, which adds four fully buffered buss slots and supplies the missing voltages. Virtually all multipaks were made before the CoCo3 came out, and need to be modified to ensure they will work properly with the CC3. There were other MPI like devices in the past, so the problem is worth a side trip....

In a 6x09 system, physical hardware is 'mapped' into the same address space as the RAM uses. Most of this space is reserved for the motherboard- the actual DRAM and the keyboard, backpanel ports, and video system. A small range is available for external devices plugged into the expansion port. External buss expanders don't know what hardware is plugged into them and their designers have to guess when a read or write is probably going to an external device- the natural assumption is any address the motherboard doesn't use must be an external hardware card. However, the CoCo3 had to first maintain compatibility with the old software, even maintaining all the old video addresses for all the old functions. The new stuff had to be placed somewhere! As a result, there are times when the GIME and the buss expander both claim the same address space. The MultiPaks can all be refitted to open up the ranges needed by the CoCo3, but the old aftermarket devices are your problem.

Even with a MPI, the problem (or it's fix) is not evident with a quick test. What happens is, the cpu attempts to reach the GIME address range, and the GIME replies. No add-on cards respond to this since it's outside their individual areas, but the buss expander turns on it's transceiver because the address is outside the old CoCo2 motherboard range and MIGHT be a valid request. Most of the time, the GIME overrides the 'dead carrier' from the expander and the system seems to work fine! Like an overly long Y-cable, you won't know until there is some really important data on the drives. Unless the unit is marked as upgraded you'll have to take it apart, or at least peer inside. What you are looking for in the old, large MPIs

is a replacement PAL chip (socketed, the original has printed Tandy copyright notice with early eighties date). The newer, small MPI doesn't have a PAL, but upgraded ones will have a small circuit board dangling from the area of it's large square IC by a half dozen wires. If neither of these mods are present, the PAL chip is probably available from Marty Goodman, while the add-on board is available (kit or installed) from CoNect. Before we leave multipaks, note that some software was hard coded to find hardware in certain slots. To avoid having to modify your software, follow this scheme: Slot1=serial port, Slot2=open, Slot3=hard drive, Slot4=floppy controller.

There was at least one CoCo3 compatible buss expansion device available, the SlotPak3. A slotpak looks like a Tandy 'long card' (old disk controller, deluxe RS232, Ore90) with three buss connectors- when plugged in, it stood on the heatsink of it's built in power supply. The added power relieved the CoCo from any heavy burden on the 5 volt line, but didn't add the missing 12v. The slots acted like a cross between a two slot MultiPak and a Y-cable- one responded as slot 1 or 3, another as slot 2 or 4, and the third was an unswitched slot. The two switched slots will accept cards with internal ROMs or incomplete address decoding (almost anything made by Tandy) while the unswitched slot accepts many 'fully decoded' aftermarket cards, notably serial ports. Note how this fits the traditional slot assignments.

Drives

Next comes the question of mass storage. Remember, OS-9 is a disk based operating system. There is going to be a lot of data exchanged in the course of a normal session, and any CoCo disk is slow by definition. You'll want the fastest, biggest drive in a hurry. A hard drive is advisable for serious use, but floppies are an acceptable starting point. And they do have a benefit- floppies are a reasonable way to get the feel of system building and filesystem handling before dumping megabytes of data onto one disk. And they are simple to install, since no multipak is needed to get the ball rolling.

While the types of floppy drives still available surplus is truly impressive, you don't want to build a library of disks that will be unreadable in 5 years! I'd stick with the current standard- a 3.5 inch, 1.44 meg floppy. These will be around new for a while yet, and while hi density isn't compatible with a CoCo floppy controller it seems all of these 1.44 meg drives come setup to react properly to the media inserted. If you buy 720K disks, it is a 720K drive and completely compatible. Very simple to install since most come with an adapter to convert from their pin style cable connector to the older cardedge used by CoCo. The same adapter kit provides the power adapter- the only real change you'll have to make is to change the drive select jumper (one of those plastic

block things) if the drive is not going to be /d1 (the _second_ floppy).

There may be one wrinkle. A FD502 floppy case uses some weird little power connectors. The adventurous (in other words, at your risk) can shave these a little to fit the standard little power connector on a 3.5 drive, if the plug is inserted 'upside down' - in other words, if you compare these two connectors side by side, with the board side of each connector oriented the same way, the 5 and 12v lines will be reversed- the flat side of the 502 connector is the notched side of the standard unit. If you are wiring in a new connector, be warned Tandy sometimes experimented a bit with the wire colors. Color to color doesn't always work. Check actual voltages before assuming anything.

In the past, the standard floppy only setup was a 360K drive 0 (for CoCo compatibility) followed with one or two 720K drives for mass storage. With the exception of DECB CoCo's, which have to deal with a few copy protected games, this setup has become a bit dated- there aren't many new releases to remain compatible to, and this scheme puts your smallest, slowest, oldest drive in the most used posi-

Do you need some of those great Tandy games, utilities, programming books, or educational software? I have copies of ROM paks and original disks with documentation for you! Replace those disk that you no longer have or are having trouble finding! Most disk software is \$10.00 (\$5 S&H). ROM Paks \$7.50 (\$3 S&H), and Cassettes \$5.00 (\$2.50 S&H). All sent UPS unless otherwise requested.

I also have a lot of educational software! It is \$5.00 for disk (\$3.50 S&H) and \$2.50 for cassettes (\$2.50 S&H). Most Tandy titles, including the "Reading is Fun" series. Children's Computer Workshop programs are \$7.00 (\$3 S&H) for disk and \$2.50 (\$2.50 S&H). Dorsett Educational Systems courses are packaged on eight cassettes. Each full course is \$7.50 (\$3.00 S&H). Write and ask for "SALES LIST FIVE" for quantities and titles. DONT'T FORGET TO INCLUDE A LARGE (#10) SASE.

I also have some hardware, including a few CoCos, disk systems, and MC-10 16K RAM modules.

Pete Bumgardner

100-D W. Falls St.

Kings Mountain, NC 28086

(704) 730-0893 (9am - 2pm EST)

tion. I'd recommend the 360K unit become the last drive, used only to read old CoCo disks. There will be disks (OS-9 games with strange boots, DECB programs with hardcoded disk routines) that practically demand a standard 158K disk- these aren't as great a problem as you might guess since the defaults work! Putting a 720K disk in your 1.44M floppy and typing DSKINI0 is a perfectly legal way to make a 158K disk. You can even use BACKUP N TO 0 (where N is the 5.25 inch drive) to generate a usable 3.5 inch disk! Even though 3/4 the disk space is wasted the little uncrunchable box your game now occupies is easier (and probably cheaper) to find than the original format.

Portage

After the floppy is going strong, a serial port is the next logical upgrade. Even if you don't yet use email and think surfing in webs sounds foolish, there is lots of software available online. Much of it isn't available any other way, and while you can download to your PC and shuffle things through the 3.5, it's far easier to download direct to CoCo. You've probably noted the serial port on the back panel. Forget about it- this minimal port is mainly suited for running a serial printer and can barely handle slow two way communication. The first useful level is a 6551 based port that plugs into the buss- these were made by dozens of vendors, but almost all of them are logically copies of Tandy's 'Deluxe RS232' pak. The main variation in serial ports lies in how they deal with the CoCo's lack of 12v power on the system buss. Tandy used an internal dc/dc converter to power standard serial line drivers, while some aftermarket units used 5v only driver chips, usually the Max232. For economy, only one is generally used, which means the port can't supply the full compliment of handshaking lines used in modem communications- this can be a problem when connecting modern, hi speed modems under OS-9 Other aftermarket ports depend on either the MultiPaks buss supplies or an external power lump to power conventional line drivers. Like the Tandy, these units will operate more or less reliably with any modem, with a few limitations.

Under stock OS-9, the Tandy/clone rig is completely reliable only at low speed- 2400bps. Even here, the cps (actual characters transmitted) is lower than the 240 expected. Faster connections often drop characters from the incoming data stream. Usable speed can be improved- first, to 9600 by changing the clock module to avoid an IRQ bug in the CoCo. The 'edition 9' series of replacement clocks (Eddie Kuns) is available from many on-line services, and will get rid of those lost characters. Another improvement comes when the cpu and opsys is upgraded to 'Nitro'. Expect 1000cps-enough to keep 19.2K open. For power modems, there is a new serial port, the Fast232. With the software clock fix, these will do

115200bps and download much faster, up to 5500 cps. The same hardware also allows more apps/users when the port is active. Unfortunately, at \$80 they cost quite a bit more than a used 'clone' port.

Hard drives

With a serial port collecting software and data, your storage needs will begin growing, and you'll want to look into a hard drive. Just like floppies, there are a variety of older setups still around, but the drives used in some of them are also starting to suffer from age. If you got a decent sized mfm drive in your scrap XT, it's worth looking for a Burke&Burke interface. This gizmo will usually accept everything from the parts XT- drive, controller card and cables. If you have to buy a drive, look for a SCSI controller from KenTon or Disto. These will control many of the current 'simple' SCSI devices (forget fast, wide, 2, or anything like that) but have one flaw- they don't supply the parity bit newer hardware is starting to require. As of 12/96 there is both an IDE and full SCSI interface under construction so this may change for 97.

Other Bits

In the category of odds and ends, you'll need a pointer. The CoCo's mouse was based on the built in joystick circuit, with an outrigger adapter to boost the resolution to 640x200. CoNect has new mice and res adapters, and used ones generally only need some cleaning to get back into shape. Open the mouse up, and you'll see plastic pickup wheels that read the movement of the steel pinball. These connect by gears to the pots (tiny box with wires) that translate the motion into electrical signals. These plastic gears are fragile, and there is a trick. The pots will swivel back and disengage the gears- just unsnap the gear side. Now you can safely clean the pickup wheels. One other mouse trick- the fancy two button mouse uses the same ball and pot assembly as the ugly black version, and TUBV is much easier to find if you need repair parts. (In fact, I've got about a dozen new ones left.)

You'll also want a printer. The main problem here is the CoCo's serial feed- a new generic s/p converter is expensive and the old CoCo specific converters are scarce. This is another thing you might be able to find in DOS surplus. On the software side, printer drivers lean towards the dinosaur level- the safest bet is anything that offers Epson emulation. Some OS-9 apps (DynaStar, Sculptor) use an ini file, allowing a wider choice of printers, but Epson text is by far the easiest.

Next month, back to the software!



Breaking the 64K Barrier!

Allen Huffman

Using pipes & paths for more memory under OS-9 Level II

One of the frustrating limits an OS-9/6809 programmer encounters is the 64K barrier imposed by the 16-bit address limitation of the CPU. Clever memory banking techniques performed by the GIME chip's memory management unit (MMU) allow access to several megabytes of system ram, but never more than 64K at a time. This "64K barrier" prevents many larger projects from successfully being attempted on a CoCo 3 running OS-9. Programming tricks can be used to get around many of these limitations, even through BASIC09.

This series of articles will primarily discuss the pipe manager of OS-9 Level 2 and how it can be used to tie separate processes together allowing you more solutions to the memory problem.

Paths - Everything's Got One

All I/O under OS-9 is done through a "path". The path is simply an integer value that corresponds to a particular "device" tied to it. At the system level, OS-9 maintains a table of paths to every instance of all active devices. From the program level, you are limited to only 16 paths at any given time. This means a program could never open more than 16 files simultaneously under OS-9. Realistically, it's more like 13 since the first three paths are typical reserved as your Standard Input, Standard Output, and Standard Error Output. (OS-9/68K and OS-9000 allow up to 32 paths at a time.)

All access to a device is done by first "opening" it by name. The device name will begin with a slash then the name of the device descriptor. For instance, a serial port descriptor might be called "/T2", so to access this port one would first open "/T2". The operating system will return a "path number" which will then be used for reading and writing. Consider the following PSUEDO CODE:

```
modem_path = open( "/t2",  
WRITE_ACCESS )
```

The operating system opens whatever device "/t2" is for write access, then returns a "path number" back. To send data to this device, you would then do something like:

```
write( modem_path, "This goes to the  
modem" )
```

When the device is no longer needed, the path is "released" back to the system by closing it:

```
close( modem_path )
```

Real code to do this might look like:

```
BASIC09:  
  
PROCEDURE main  
DIM path:INTEGER  
  
OPEN #path,"/T2":WRITE  
PRINT #path,"This goes to the mo-  
dem."  
CLOSE #path  
END  
  
'C':  
#include <stdio.h>  
#include <modes.h>  
  
main()  
{  
    int path;  
  
    path = open( "/T2", S_IWRITE );  
    writeln( path, "This goes to the  
modem\n", 80 );  
    close( path );  
}
```

But what exactly is a path? First, it must be realized that three paths are automatically assigned to every process that is created under OS-9. These paths are known as "Standard Input", "Standard Output", and "Standard Error Output". Suppose you are using a window device called "/W5" to run a program on. When it displays a message, it goes to the "Standard Output" path which is /W5. When it reads data from the user, this is read from "Standard Input" which is also /W5. Error messages are traditionally printed to the "Standard Error Output" path which, in this case, is also /W5. For most situations, input, output and error output will all go to the same path.

Each process has a "path table" which holds these paths, and any others the program opens. The first three slots are de-

finied to be STDIN, STDOUT, and STDERR. I will list them in uppercase so they don't get confused with the "FILE* stream" structures used by the 'C' language. A sample path table for a process might look like this:

Process A - path table upon process creation

```
0 - /w5  STDIN  
1 - /w5  STDOUT  
2 - /w5  STDERR  
3 - <available>  
4 - <available>  
...  
14- <available>  
15- <available>
```

Default "rules" are used by languages such as 'C' and BASIC09 to send I/O to the correct place. For instance, doing a "PRINT" in Basic09 actually does a WRITE to path #1 by default. In 'C', the "printf" function uses a "file stream" called "stdout" which, at the operating system level, is using path #1 as well. They both end up using the same kernel I/O functions.

You may notice that, in order to read input from the keyboard under BASIC09, you must do something like "READ #0,A\$" or "INPUT #0,A\$". This is instructing it to read from path #0, whatever that might be (terminal, window, etc.). What it is doesn't concern the application. All it knows is that paths 0, 1, and 2 have (hopefully) been opened by something earlier and made available. But what does "open" actually imply?

If this process opens a path to, for instance, the serial port, it will be assigned the LOWEST available path slot. The following code:

```
OPEN #modem,"/T2":WRITE
```

...would make the path table look as follows:

Process A - path table after opening the serial port

```
0 - /w5  STDIN (read)  
1 - /w5  STDOUT (write)  
2 - /w5  STDERR (write)  
3 - /T2  (write access)  
4 - <available>
```

```
...
14- <available>
15- <available>
```

The path variable, "modem", would contain a value of 3 indicating that path slot 3 is "connected" to the serial port just opened. So, at this point, doing either of the following:

```
PRINT #modem,"This goes to the mo-
dem"
or
PRINT #3,"This goes to the modem"
```

...would have the same effect. This behaviour is predictable, but the return path variable should always be used to ensure you are using the correct path slot the operating system actually returned.

The Shell

About the only time we see pipes come into play under OS-9 Level 2 is when they are used from the shell. The "shell" is simply a program that knows how to "parse" what you type and do certain things. For instance, if it sees something it doesn't recognize, it tries to "fork" it as a process or run it as a script file. Several special characters can be used to alter the three standard paths. The are "<", ">", and ">>" which change STDIN, STDOUT, and STDERR to go somewhere else.

To send something to the printer, you have perhaps seen the following shell command line:

```
OS9: list file.txt >/p
```

This literally means "fork the utility called 'list' passing it the parameter 'file.txt' and redirect it's standard output to a device called '/p'". When this command is executed, the "list" process' path table would look like:

Process 'list' - path table upon process creation with output redirected

```
0 - /xx  STDIN (whatever it is)
1 - /p  STDOUT (now redirected
to go to the "/p" device)
2 - /xx  STDERR (whatever it is)
3 - <available>
...
```

The program "list" has no idea that it's output is going somewhere other than your terminal screen. It simply, blindly dumps output to "path 1", the defined "standard

output" path. Using "<" redirects STDIN in a similar manner, and ">>" redirect STDERR. Using them all together like this:

```
OS9: program <>>>/device
```

...launches this "program" with it's path table setup so path 0, 1, and 2 all go to "/device", whatever that is. The only question is, what does the shell do with the original paths and how does it get them back?

I/O Redirection

If you blindly close a path, such as the output path #1, you are in trouble. Your application will no longer have an outlet for displaying messages. PRINT and printf will fail. (How many 'C' programmers ever bother checking the return value of printf to see if it really worked?) This would be bad. As noted above, though, the only way to make that path available so you can "redirect" it somewhere else is to close it. So, to safely do this we must first "save" the path by storing it in an un-used path slot, then we can close the original and re-open it somewhere else.

This is done by an OS-9 function known as "ISDup". It merely "clones" an existing path and copies it somewhere else in the path table, then returns the location it copied to. In 'C' it would look something like this:

```
'C':
main()
{
    int outpath;

    outpath = dup( 1 );

    /* See comments below */

    close( outpath );
}
```

For BASIC09 it gets a bit messier since there is, apparently, NO support for the "dup" system call. There are two approaches that work. I was able to write a small 81 byte "dup" subroutine function that is callable from BASIC09, as well as using the syscall subroutine to do the same thing. Both options will be presented here, but the end result is the same.

```
BASIC09:
PROCEDURE duptest
```

```
DIM outpath:integer
RUN dup( 1, outpath )

/* and so on... */
```

```
CLOSE #outpath
END
```

...or...

```
PROCEDURE duptest2
```

```
TYPE
REGISTERS=CC,A,B,DP:BYTE;
X,Y,U:INTEGER
```

```
DIM regs:REGISTERS
DIM code:BYTE
DIM outpath:integer
```

```
REM * register A=path to dup
regs.A=1
```

```
REM * $82 = OS9 function for ISDup
RUN syscall( $82, regs )
```

```
REM * here, regs.A has the new path
outpath=regs.A
```

```
/* and so on... */
```

```
CLOSE #outpath
END
```

When this program begins, it's path table is 0-IN, 1-OUT, and 2-ERR. When path 1 is "duplicated" in this manner, it gets copies into the lowest available path slot, which is 3. The path table would now look like 0-IN, 1-OUT, 2-ERR, and 3-OUT (a copy of the original). At this point, writing to either path 1 or 3 would have the same effect. Why do this? Because, with the paths setup in this manner, the program could then do something like this:

```
main()
{
    int outpath;
    int modem;

    /* save our current output path */
    outpath = dup( 1 );

    /* close our original output slot */
    close( 1 );

    /* open a path to the modem */
    modem = open( "/T2", S_IWRITE );
    /* "modem" comes back as 1, */
    /*since it takes that slot */
```

HawkSoft

28456 S.R. 2, New Carlisle, IN 46552
219-654-7080 evs & ends MO, Check, COD; US Funds
Shipping included for US, Canada, & Mexico

MM/1 Products (OS-9/68000)

CDF \$50.00 - CD-ROM File Manager! Unlock a wealth of files on CD with the MM/1! Read most text and some graphics from MS-DOS type CDs.

VCDP \$50.00 - New Virtual CD Player allows you to play audio CDs on your MM/1! Graphical interface emulates a physical CD player. Requires SCSI interface and NEC CD-ROM drive.

KLOCK \$20.00 - Optional Cuckoo on the hour and half hour!! Continuously displays the digital time and date on the /term screen or on all open screens. Requires I/O board, I/O cable, audio cable, and speakers.

WAVES vr 1.5 \$30.00 - Now supports 8SVX and WAV files. Allows you to save and play all or any part of a sound file. Merge files or split into pieces. Record, edit, and save files; change playback/record speed. Convert mono to stereo and vice-versa! Record and play requires I/O board, cable, and audio equipment.

MM/1 SOUND CABLE \$10.00 - Connects MM/1 sound port to stereo equipment for recording and playback.

GNOP \$5.00 - Award winning version of PONG(tm) exclusively for the MM/1. You'll fo crazytrying to beat the clock and keep that @\$%& ball in line! Professional pongists everywhere swear by (at) it! Requires MM/1, mouse, and lots of patience.

CoCo Products (DECB)

HOME CONTROL \$20.00 - Put your old TRS-80 Color Computer Plug n' Power controller back on the job with your CoCo3! Control up to 256 modules, 99 events! Compatible with X-10 modules.

HI & LO RES JOYSTICK ADAPTER \$27.00 - Tandy Hi-Res adapter or no adapter at the flick of a switch! No more plug and unplugging of the joystick!

KEYBOARD CABLE \$25.00 - Five foot extender cable for CoCo 2 and 3. Custom lengths available.

MYDOS \$15.00 - Customizable, EPROMable DECB enhancement. The commands and options Tandy left out! Supports double sided and 40 track drives, 6ms disk access, set CMP or RGB palettes on power-up, come up in any screen size, Speech and Sound Cartridge support, point and click mouse directory, and MORE OPTIONS than you can shake a stick at! Requires CoCo3 and DECB 2.1.

DOMINATION \$18.00 - Multi-Player strategy game. Battle other players armies to take control of the planet. Play on a hi-res map. Become a Planet-Lord today! Requires CoCo3, disk drive, and joystick or mouse.

SMALL GRAFX ETC.

"Y" and "TRI" cables. Special 40 pin male/female end connectors,

priced EACH CONNECTOR - \$6.50

Rainbow 40 wire ribbon cable, per foot - \$1.00

Hitachi 63B09E CPU and socket - \$13.00

512K Upgrades, with RAM chips - \$72.00

MPI Upgrades for all small MPIs (satellite board) - \$10.00

Serial to Parallel Converter with 64K buffer, cables,
and external power supply - **NOW ONLY \$28.00!!!**

Serial to Parallel Converter (no buffer), cables,
and external power supply - **ONLY \$18.00!!!**

2400 baud Hayes compatible external modems - \$15.00

Modem cable (4 pin to 25 pin) - \$5.00

ADD \$3.00 S&H FOR FIRST ITEM, \$1.00 EACH ADDITIONAL ITEM

SERVICE, PARTS, & HARD TO FIND SOFTWARE WITH COMPLETE DOCUMENTATION AVAILABLE. INKS & REFILL KITS FOR CGP-220, CANON, & HP INK JET PRINTERS, RIBBONS & vr. 6 EPROM FOR CGP-220 PRINTER (BOLD MODE), CUSTOM COLOR PRINTING.

Terry Laraway
41 N.W. Doncee Drive
Bremerton, WA 98311
360-692-5374

**NOTE NEW ZIP CODE
AND PHONE NUMBER!**

The BlackHawk MM/1b

Based on the AT306 board from Kreider Electronics. Features built into the motherboard include:

16 bit PC/AT I/O bus with five slots

MC68306 CPU at 16.67MHz

512K to 16MB of RAM with

30 pin SIMMs (4 sockets)

IDE Hard Drive Interface (2 drives)

360K-1.44MB Floppy Drive

Interface (2 drives)

Two 16 byte fast serial ports

(up to 115K baud)

Bi-directional parallel printer port

Real-time clock

PC/AT keyboard interface

Standard PC/AT power connector

Baby AT size - fits standard PC case

BASIC (resembles Microsoft

BASIC)

MGR Graphical Windowing Envi-

ronment with full documentation

"Personal" OS-9/68000 Vr 3.0

(Industrial with RBF)

Drivers for Tseng W32i and

Trident 8900 VGA cards

Drivers for Future Domain 1680 and

Adaptec AAH15xx SCSI cards

OS-9/68000 Vr 2.4 with Microware

C 3.2, Assembler, MW Basic (like

Basic09), MW Debug, MW Pro-

grammers Toolkit

UUCP from Bon Billson

Ghostscript (software PostScript

interpreter)

Many other utilities and tools

Prices start at \$400!

(motherboard,

Personal OSK, & MGR only)



BlackHawk
Enterprises, Inc.

756 Gause Blvd. #29

Slidell, LA 70458

E-mail: nimitz@stolY.com

```
/* from this point on, anything that uses
   path 1, the standard output path,
   will be going to the modem, so... */
```

```
printf( "Hello. This now goes to the
modem.\n" );
```

```
/* when done, we must set things back as
they were. */
```

```
/* we close the modem path (1) so it will
again be available */
close( modem );
```

```
/* then we "dup" our saved out path back
into it's original slot */
dup( outpath );
```

```
/* we don't bother finding out where it
was put since we don't care */
```

```
/* finally, we close our saved copy */
close( outpath );
}
```

If this seems confusing, read it over a few times and try to envision what the path table looks like between each step. This might help:

```
Start: 0-IN, 1-OUT, 2-ERR
Dup: 0-IN, 1-OUT, 2-ERR, 3-OUT
(copy)
Close: 0-IN, <...>, 2-ERR, 3-OUT
Open: 0-IN, 1-/T2, 2-ERR, 3-OUT
...
Close: 0-IN, <...>, 2-ERR, 3-OUT
Dup: 0-IN, 1-OUT, 2-ERR, 3-OUT
Close: 0-IN, 1-OUT, 2-ERR
```

One might immediately ask "what's the point?" since it seems easier for the program just to open a new path and do all the reading and writing to that device. Sure, one could have easily done:

```
/* using high-level buffered I/O */
main()
{
    FILE *modem;

    modem = fopen( "/T2", "w" );

    fprintf( modem, "Hello. This now
goes to the modem.\n" );

    fclose( modem );
}
...or...
/*using raw low-level non-buffered I/O */
main()
{
```

```
int modem;

modem = open ( "/T2", S_IWRITE );

write( modem, "Hello. This now
goes to the modem.\n", 80 );

close( modem );
}
...or...
REM * BASIC09 version

DIM modem:INTEGER

OPEN #modem,"/T2":WRITE
PRINT #modem,"Hello. This now
goes to the modem"
CLOSE #modem
```

If you have access to the source code, and don't mind putting in all the extra path references, this is fine. But what if you don't have the code to all the library routines the function may call? Such as:

```
main()
{
    /* your code */
    somefunc(); /* library function */
    /* your code */
}
```

In this example, regardless of the path you opened in main, the library function may still be printing to stdout or writing to path 1, unaware of anything else you might like to do. For these situations, redirecting your own paths is a simple solution.

But wait... It gets better. Suppose you have a completely separate program you might like to redirect. If you don't have the source code, then recompiling it to always use the modem or printer isn't an option. You could always run it from the shell and use the ">" redirectional modifiers, OR you can do it yourself under program control. Here's how it works:

When a program "forks" a new "child" process, the new process gets a COPY of the first three paths the "parent" is using. The child can close it's copies without affecting the parent. PSUEDO code might look like this:

```
parent makes it's paths look like: 1-
IN, 2-printer, 3-ERR
parent forks a child so:
child's paths look like: 1-IN, 2-
printer, 3-ERR
```

while child runs, all printing to STDOUT goes to the printer parent can restore it's paths back, without affecting the child

Some code might look like this:

```
/* same as above, but one important
difference */
main()
{
    int outpath;
    int modem;

    /* save our current output path */
    outpath = dup( 1 );

    /* close our original output slot */
    close( 1 );
```

```
/* open a path to the modem */
modem = open( "/T2", S_IWRITE
);
/* "modem" comes back as 1, since
it takes that slot */
```

```
/* IMPORTANT DIFFERENCE!
Here, we fork a child process so
it starts life with a copy of our
newly redirected output */
```

```
system( "dir" );
```

```
/* when done, we must set things
back as they were. */
/* we close the modem path (1) so it
will again be available */
close( modem );
```

```
/* then we "dup" our saved output
path back into it's original slot */
dup( outpath );
/* we don't bother finding out where
it was put since we don't care */
```

```
/* finally, we close our saved copy */
close( outpath );
}
```

The net result? We have a program that will always launch the "dir" command (or whatever) with it's output going to the modem. Too much work? Perhaps. Some of you might already be saying that it's just as easy to let the shell do it like this:

```
system( "dir >/t2" );
```

And, of course, you are right. Again, we are back to the "why bother" stage. I

suppose it's important to explain this just to show you HOW it could be done, not tell you WHY one should do it that way. If you recall, this entire topic started out about pipes, which haven't been mentioned since. So, let's mention them.

Pipes and other Unmentionables

The shell provides a way to "tie" one programs output into another programs input from the command line. For instance:

```
OS9: prog1 | prog2
```

Doing this makes everything that "prog1" would normally print to the screen be read by "prog2" as if it were standard input. Suppose you wrote a program that just read lines from standard input until there were no more, counting them. When done, it would display how many lines it found. You could then use that along with a "pipe" like this:

```
OS9: list file.txt | countlines
```

The "countlines" program starts reading from standard input. Running it by itself would not accomplish much since you'd be forced to type everything in from the keyboard. BUT, in the above example, we are letting the "list" utility dump the contents of a file INTO the input of "countlines". End result? An easy shell utility to count how many lines are in a file, without ever having to write any code that physically OPENS the file and counts them.

How does this work? Well, the shell pulls it's own tricks. If you look in the OS-9 Technical Reference manual on page 7-2 you will see a sample logic flow of how the shell does this. Again, what does it all mean?

There is a psuedo device (i.e., no hardware) called "/PIPE" which doesn't even have a device drive (no hardware, remember?). It uses a file manager called "PipeMan" and acts as sort of a virtual "serial buffer". You can open a path to "/pipe" and write data into it, then read it out. Not very useful, but it can be used to tie multiple processes together and let them exchange common data.

The problem here is simple. How do the programs all access the same pipe? Every time you open "/pipe" you get a brand new un-named pipe. Under OS-9/68000 and OS-9000, "named" pipes are available

which make the job much easier. (Joel Hegberg has a fine tutorial on these in past issues of this publication.) Every process that wants to share data can just open something like "/pipe/whatever" then they can all read and write and be talking. Unfortunately, named pipes were never _released_ for OS-9 Level 2. The ONLY way to legally share an un-named pipe is by "passing" it's path to a child process. Since a child process only "inherits" the first three paths, the parent MUST put any "pipes" to be passed down into one of those slots before launching the child. Let's look at some psuedo code again:

```
parent makes it's paths look like: 0-IN, 1-OUT, 2-/pipe
```

```
parent launches a child:
```

```
child's paths look like: 0-IN, 1-OUT, 2-/pipe
```

if the child "writes" into path 2, the parent can "read" that data from it's own path 2.

This limits the amount of "normal" I/O either process can do since both of their standard error output paths are now tied up by the pipe. A "better" approach might be to only use those paths just for transferring the pipe from parent to child, then restoring the paths back like they should be. Unfortunately, under OS-9 Level 2 only three paths can be passed down so one path will be "lost" for each pipe you want the child to know about.

Is this a problem? Not really. Let's look at how this might be used. Suppose you are writing the next "killer app" for OS-9 and you've finally run out of room in the 64K address space. You could go out and buy a new AT306 OS-9/68000 computer and port it over to a 32-bit system, OR consider the possibilities pipes might offer for...

Breaking the 64K Barrier using Pipes

Just as a good television series might end with a cliffhanger during the ratings sweep, I close this discussion but will, if there is interest, continue it in the future. Please write in and let the publisher know if articles like these would be of interest. Until that time, crack open your OS-9 Technical Manual and explore pipes for yourself.



What are you waiting for?

Get your friends to subscribe to the only magazine that still supports the Tandy Color Computer...
"the world of 68' micros"
The more people who want the support, the longer it will be here!

Announcing Nitro Level III!

How many times have you been unable to load a driver due to not enough system RAM? How would you like to have up to 32K of system RAM available? How is this possible?

In effect, Nitro Level III turns the system into a Kernel only process, with 48K of RAM, and 2 IO processes (RBF and SCF), each with 16K of RAM. This is similar to Grfdrv having it's own 64k memory area.

The kernel process contains the minimum modules to run an OS-9 system, and also the descriptors. The RBF/SCF processes contain the IO modules, and the IO buffers.

There are 2 big benefits here:

1 - Both RBF and SCF are not in system memory at the same time, so you save RAM.

2 - You don't have 16K of SCF or RBF modules, so everything up to 16K can be used as device data storage (sector buffers, etc.)

Level III works only with Nitro (all versions). It can be purchased from FARNA Systems alone (\$20) or with the latest version of Nitro (\$45 for Nitro v2.00 and Level III). See the FARNA ad in this issue for ordering information.

Block Copies...moving stacks of memory around.

Shortly after the CoCo III was released, I was amazed to see programs such as ColorMax III move such a large amount of memory around so quickly. I kept thinking to myself, so what's the trick? Some hidden instruction I don't know about?

So I sat down, grabbed by 6809 quick reference card and went to work looking for the answer. It didn't take long before it hit me, the only instruction that can load a large number of bytes from memory are those that deal with the stack!

Since the PSHS/PSHU/PULS/PULU instructions can work on all registers (except themselves) it only made sense that that was what was being done. The following example shows how to move 7 bytes from one location to another.

```

PULS  Y,X,D,DP
* (13/12) Get 7 bytes
PSHU  DP,D,X,Y
* (13/12) Store 7 bytes
LEAU  14,U
* (5/5) Adjust to next dest pointer
    
```

You'll notice the numbers in parentheses. These are the number of cycles each instruction takes on the 6809 (the first number) and the 6309 (second number) when running in sensitive mode. Now take a look at the following code:

```

LDD  ,X++
* (8/7) Get 2 bytes
STD  ,Y++
* (8/7) Store 2 bytes
LDD  ,X++
* (8/7) Get 2 bytes
STD  ,Y++
* (8/7) Store 2 bytes
LDD  ,X++
* (8/7) Get 2 bytes
STD  ,Y++
* (8/7) Store 2 bytes
LDA  ,X+
* (7/6) Get a byte
STA  ,Y+
* (7/6) Store a byte
    
```

When we add all of the timings up, we come up with a 100% speed increase if we use the stack to move data around. That means that you could move 2 32k screens around in about a second instead of 1 screen in a over a second. Big difference!

But, there are some things to consider when doing this. Using the stacks to move data around means that you cannot have the interrupts enabled. This is because anytime an IRQ or FIRQ happens, register contents are stored on the system stack (S register). This means

that important data can be overwritten if an interrupt occurs. Unless you move only 6 bytes at a time, changing the contents of the DP register can be very bad if you do not restore it when complete.

The following routine handles those and other minor problems that can be encountered.

```

*****
* Move an 8k chunk of memory from one
location to another *
* Entry: *
* X Source location where the
data is to be moved from *
* U Destination location the
data will be moved to *
* *
* All registers are preserved in this call
*
    
```

```

*****
SSTACK  RMB  2
Area to save system stack
COUNT  RMB  1
Counter for number of passes
STKMOV  PSHS
D,X,Y,U,CC,DP Save all registers
ORCC  #$50
Disable interrupts
STS  SSTACK Save
the current system stack
LDA  #130 Move
63 bytes 130 times
STA  COUNT
Save it
TFR  X,S Move
source into S register
LEAU  7,U
Adjust destination pointer
* Move 63 bytes
SMOVB PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
    
```

```

PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
PULS  DP,D,X,Y
Get 7 bytes
PSHU  DP,D,X,Y
Store 7 bytes
LEAU  14,U
Adjust dest point 14 bytes
DEC  >COUNT
Decrement counter
BNE  SMOV0
Keep going until we are done
* We have just moved 8190 bytes, and we
have 2 left
LDD  ,S
Get the last 2 bytes
STD  -7,U
And save them
LDS  SSTACK R e -
store the system stack
* Restore registers, IRQ/FIRQ status and
return
PULS  D,X,Y,U,CC,DP,PC
    
```

Since this routine moves around 8k at a time, it is perfect for the CoCo III since the MMU works in 8k chunks. I have used routines like this in several things including Image Master, Blox and Digger. You can also deviate from this and use the same technique for clearing large portions of memory at a time.

Next time, we will start writing a small program that plays digital music similar to .MOD files, but does not require the overhead and can play both 2 channel and 4 channel (Using the Orchestra-90 pak) music.



Modify your RS-232PAK

John Kowalski

Stop losing characters!

I've been running a BBS on my CoCo for the last 11 years or so. When I finally got myself a 14.4Kbps modem a year ago, I decided to simply run my RS-232 at the maximum rate (19200 baud) and leave it locked there. No matter what baud someone called in at, the flow control lines are supposed to keep my BBS from sending out too much data to low speed callers.

Well, it turns out that the 6551 has a lousy way of dealing with CTS line changes. Every time the modem would raise the CTS, the 6551 would stop dead in it's tracks - right in the middle of sending a byte! For low speed callers, that meant every 200th byte or so was truncated and perceived as trash.

I solved this problem the easiest way I could. I removed the CTS signal (pin 9) from the 6551 altogether and rewired it to the cassette input (pin 4 on cassette port) with a 22K Ohm resistor. I then forced the 6551 to always think the modem is ready by grounding the CTS on the chip (Solder a wire from pin 9 to pin 1 on the 6551). I used the cassette port because it's a free input port that's generally unused by everyone. This works, but you have to patch your RS-232 driver to accommodate it. My new driver simply checks the state of the cassette port before deciding if it should wait to transmit the next byte or not. Even with a stock driver the 6551 will no longer destroy outgoing data. No problems will arise except in the case of the computer sending out more than 1.5K or 2K ahead of the modem, which is unlikely. This is simple enough not to need a diagram, so I didn't include this in my drawing.

Now for what IS in the diagram. Normally, the 6551 simply has no way of telling my mo-

dem when it's data buffer is full. If someone called my BBS at a high baud rate and tried uploading a text file, my driver simply didn't read the data fast enough to get it all without missing bytes. Every time the disk drive went on, the CPU halts and the 6551 misses more data. In order not to miss data, the RS-232 driver would have to immediately read data from the 6551 every time a byte came in. The solution? I made a little circuit that sits on top of the 6551 chip (a 74ls32 and 74ls00) that generates a new RTS control signal. Since modern modems have a data buffer, it would be really neat to unload a lot of work from the CoCo and leave it for the modem to do. Every time the 6551 receives data, the circuit raises the RTS line (which tells the modem to stop sending to the CoCo) until that data is read by the software. If the driver reads the data at the full incoming speed, there is no slowdown in throughput. If the driver goes on a break, or the disk drive comes into use, the modem is told not to send any more data until the 6551 is ready to receive it!

Effectively, the circuit disallows the 6551 from missing any data. It also places the job of buffering incoming data on the modem, and not the CoCo. The CoCo doesn't have to keep reading the 6551 every time a byte comes in, now it can read the data when it actually wants it. I only read from the 6551 when the BBS wants input from the user. If the user transmits stuff while the BBS is calculating, loading, thinking, etc... the modem will buffer it for me, speeding things up quite a bit.

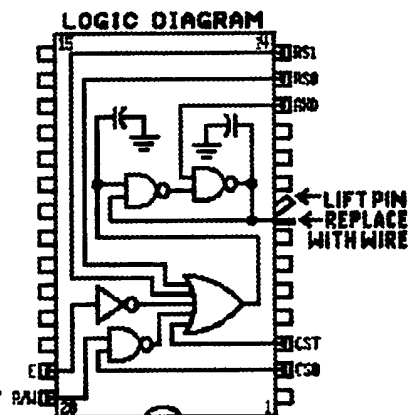
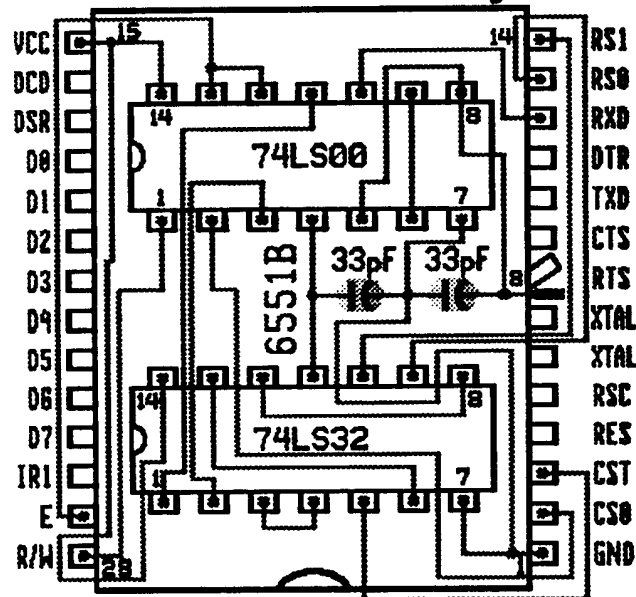
Note: I've recently been informed that the RTS modification has a side effect when used

with certain kinds of modems. It works wonderfully with my U.S. Robotics 14.4 Sportster which has a 2K data buffer. But, it may function inefficiently with other modems lacking data buffers. When the RTS coming from the the CoCo signals the modem that the 6551 is not ready to receive data, these modems may pass along this signal to the HOST computer and tell it to stop sending data until the CoCo is ready for more. This happens because the modem does not buffer incoming data, therefore cannot store it if the CoCo is not ready to receive. This extra step adds a delay between incoming characters because the signal ends up travelling huge distances across the phone line between each character.

I also rewired the line that goes to Carrier Detect (DCD pin 16 on 6551) signal to go to the DSR (pin 17) on the 6551. Make sure to cut any lines that previously went to the DSR pin on the 6551. Then, I wired the DCD (pin 16) to ground (pin 1). This way I can still receive data when there is no carrier, but I can still be able to detect carrier status by checking the DSR state. (This is not shown in the diagram either because it's as easy as cutting two wires and rewiring two pins.)

Lastly, I replaced the 1.842Mhz clock crystal on the 6551 with a 3.684Mhz one. This doubles all baud rates. The 6551 can now run at 38400 bps! Many terminal programs that use GIME timer interrupts to drive the RS-232 will still sample the RS-232 at their old sampling rate (half the new rate required) which effectively reduces your throughput or causes lost data. So this modification is not recommended unless you know that your software is driven by the 6551 Interrupt line.

Sock Master's RS-232 RTS Flow Control Modification Circuit Diagram



Sock Master's Hardware Handshaking RTS for 6551. This circuit replaces the old RTS with a new signal that tells the modem to stop sending data when the RS232 buffer is full. This stops the RS232 from EVER missing incoming data!

These four modifications each serve a purpose, but you don't need to attempt them all. Each particular modification will do it's job all by itself. You can try getting started with the easy modifications to the DCD or CTS first and then decide if you want to try the more serious hardware hacking of the RTS modification. If you want just a particular function of the 6551 'fixed', then you can do just that modification too. I have all four done to my RS-232, and they work wonders for the operation of my BBS.



Messages....

continued from page 3

Dear Frank,

I'd like to pass on an internet resource that I think would be very beneficial to our community. It's a service called "pobox.com". They provide e-mail aliasing. Basically, you sign up with them and they give you an e-mail address on their system which will be forwarded to whatever real server you use. Thus, if you send mail to me as "allenh@pobox.com", the letter will be forwarded to my real account at "cocosysop@genie.geis.com". It makes remembering e-mail addresses much easier.

They give each member a three month FREE trial, then there is a yearly fee of \$15. With this, you get three e-mail aliases. For instance, I can be reached as:

allenh@pobox.com
allen.huffman@pobox.com
subetha@pobox.com

All three go to the same place. If I change e-mail providers down the line, I just update the forwarding address with pobox.com. They also let web pages be aliased. My real webpage is currently:

www.geocities.com/siliconvalley/184

Most websites are not very easy to remember. But, thanks to pobox.com, you can find my homepage as:

www.pobox.com/~allenh
www.pobox.com/~allen.huffman
www.pobox.com/~subetha

I think it would be great if more of our Community members with e-mail access would try out pobox.com. Then, we might one day be able to contact vendors like: connect@pobox.com, hawksoft@pobox.com, 268micros@pobox.com, farna.systems@pobox.com, wittman@pobox.com, etc...

Wouldn't it be nice?

Allen Huffman
(not affiliated with pobox.com - just a very happy user)

Glad to pass the info on Allen! Maybe some of our readers will need something like this.

Hi Frank,

Got the Jan/Feb issue a week ago. I don't think you should get involved in the PC world except to the extent of helping people use 68xxx and 68xx emulators. There's too many other people going broke doing that.

I just bought Jeff Vavasour's CoCo3 emulator. It's definitely worth \$25, if only for the entertainment value of seeing a 25MHz processor choking on OS-9-6809 software. Well, it does let me use OS-9 on my Digital notebook. But

the documentation is sparse, especially for OS-9 use. Don't misunderstand me. I think Jeff has done us all a service. But it is much more of a DECB emulator than an OS-9 emulator. Cyrus runs twice as fast as my CoCo3, while OS-9 is at best half the speed. Disk access is painfully slow compared to a CoCo3 with a hard drive. The keyboard routine loses the occasional character if you're working more than one process. If I get it running more or less reliably I'll give you a report.

On a more positive note, your (ever diminishing) advertizers are getting my attention. CoNect just shipped my Fast232, Jeff Vavasour is \$25 richer, and I'm just ordering Dennis Kitz's book and some hardware from Terry Laraway.

Have you got the AT306 running? I'm waiting for your report on how it works, and whether a 6809 emulator is available. I hate to waste that \$300 Sculptor database, not to mention Wordstar and Dynacalc.

Now here's a response to Vern Larson in this issue. I lived near the Pacific Ocean for several years and suffered through card edge connector cleanings every three months. I used the art gum method. It worked every time, every time, every time....Finally I did some research and ordered a product called Cramolin from Caig Laboratories in San Diego. Two tiny bottles. One a de-oxidizer and the other a preservative. I haven't had to use that art gum since 1992. It's expensive, so don't buy the industrial size 10 oz. can. The incredibly tiny bottles will last your lifetime.

Ion Michaelides
ion@unisono.ciateq.mx

Thanks for the tips on contact cleaner and preservative! I have no intentions of doing any more than offering tips on running the CoCo emulators on PC based machines.

We have had some troubles with the AT306 of late (see article in this issue), but it appears we have the bugs licked. I'll find out in a week or so. There are some 6809 emulators for OS-9, but these aren't CoCo emulators. I'm not sure if there are any CoCo specific emulators for 68K processors. One problem would be the speed. It takes a LOT of horsepower to emulate the code of one microprocessor on another. The 6809 emulators that do run on 68000 processors aren't very fast. If I find a decent 6809 emulator I'll let everyone know!

A Letter from Tandy!

Dear Mr. Swygert

Thank you so much for your newsletter! And yes, I was mistaken, Tandy does have a small selection of Coco software that's offered through Radio shack Unlimited.

At any rate, your newsletter was very informative. I was so pleased to find current addresses for BlackHawk and CoNect! I had written to them also, only to have my letters returned because the forwarding order had expired.

If you know of any other vendors or user's groups that still provide information to new CoCo

AT306 Trials...

continued from page 7

Well, Carl isn't sure what happened. He tested the last batch of boards, found a few with correctable problems, and sent them back for repairs. Since they had already been tested once, and Kevin Pease (the person making the repairs) tested them, Carl sent them on out without further testing... the only set of boards to leave Kreider Electronics without his personal checking, and the only set to have a flaw!

How did this happen, you might ask? The boards were tested to ensure the integrity of the repairs, right? This is a small operation, but not incompetent. Kevin Pease tests boards with a terminal connected to /t0. This means to test boards with the on-board ROM DEBUG all he has to do is attach a power supply and terminal, not install it in a case. From now on, Carl is going to test each with a video card as well!

So where are we now? Carl is having a new set of PALs burned, and is making sure that I get one early next week. My customer with the other board is receiving one directly from Carl also.

Since it seems the video card problem is taken care of, we should be up and running shortly. By the next issue I'll be writing an article for software setup and installation on a running system. Unless Mr. Murphy isn't through with us yet...

users, I would truly appreciate it if you would let me know. I always want to provide our customers with a reliable listing.

Yolanda Orozco - Correspondent
Tandy Retail Services Product Support Center
200 Taylor Street, Suite #600
Fort Worth, TX 76102

I had received a letter from Mrs. Orozco about a month prior to this one. She was asking if I still supported the CoCo. I sent a copy of 268'm and a letter in reply. I pointed out to her that Tandy still had some software available... apparently she was unaware of this, which is no big surprise.

I realize that many of you feel Tandy abandoned us when they dropped the Color Computer. Many people have called or written Fort Worth over the years, however, and always get a reply with a list of current CoCo emulators. Tandy may not be making the CoCo anymore, but they aren't nudging users out in the cold either! If you have a local group, drop Yolanda a note and let her know!

BMPREADR & PCXREADR

Stuart Wyss-Gallifent

View MS-Windows ".BMP" & ".PCX" format pictures on your CoCo 3.

PCX READER and BMP READER are two programs for the CoCo 3. They will both read Windows graphic files and display them on the CoCo 3 graphics screen. There are pros and cons to each program, which will be explained in subsequent paragraphs. The programs require a 128k or 512k CoCo 3 or CoCo 3 Emulator, and a single disk drive. The programs are 2 granules each, and written in BASIC. The BASIC program pokes in some machine-language codes for the actual loading of the Windows file.

PCXREADR.BAS and BMPREADR.BAS are both run from the OK prompt in BASIC. Type either RUN"PCXREADR" or RUN"BMPREADR" and press Enter. The program will initialize the machine-language, and then ask you for the drive where the pictures are. If you have just one drive, or your picture files are on drive 0, just press ENTER. Then you will be asked for a filename. If you just press ENTER, you'll get a directory listing of the PCX or BMP files on the drive. After typing in the filename, the program will check that the file is supported, and then it will load. After displaying, press any key to exit to BASIC. If you are using BMPREADR, after displaying the file, press the P key to select from three different palette adjustments.

The PCX Reader program will load Windows PCX files. Windows must be in 256 color mode (not 1024 or 16.7) and the picture size must be less than or equal to 320x192. In Windows Paintbrush, you can set the attributes of the picture. Click on "pels" to see the actual pixel size of the image. You can also change the color palette to anything you want. Now, draw your picture using Paintbrush. When you save it, make sure you change the format to PCX. The picture should load right into the CoCo 3.

The BMP Reader has less limitations

that the PCX Reader. Windows can be in any color mode (VGA, SVGA, etc.). Set the pixel (pel) size to less than or equal to 320x192. You can change any of the palettes. When saving, change the format to BMP (256 color). Since the CoCo 3 has a palette of 64 colors, some quality loss will occur when reducing 256 colors to 64. After loading the picture with BMPREADR, use the P key to switch among three methods of reducing the colors. Because of this loss, free-hand graphics look better than photo-quality pictures.

With either program, you are not limited to Windows Paintbrush. Any Windows program that creates or modifies PCX or BMP files can be used. I have used HiJaak Pro successfully. Remember that the file must be within the limitations of either the PCX or BMP Reader.

I use the PORT program provided with the CoCo 3 Emulator to move files from DOS to a CoCo disk. Other programs are available. When using PORT, do not add or strip linefeeds or carriage returns, and select the file types to "Data" or "Exec", and "Binary".

The file on the CoCo disk should be exactly the same size as on the PC. Other transfer programs may have similar settings. If the resulting file is not readable by PCX or BMP Reader, the program will tell you what is wrong, and stop loading the file.

The author can be contacted at the following addresses: Stuart Wyss-Gallifent, 2123 Longview Road, Warrington, PA 18976-1524 swyss@voicenet.com, or my homepage (for these and other programs) at <http://www.voicenet.com/~swyss>

```
1 'BMP READER (C)1996 BY STUART
  WYSS-GALLIFENT SEPT 9, 1996
2 CLEAR2000,&H5FFF:DIM T(50),SS(50),SK
  (25):LE=0:HI=0:POKE&HE8C8,33
3 POKE65497,0:PRINT"INITIALIZING...":
  GOSUB54
4 WIDTH80:PALETTE0,0:PALETTE8,63:
  ATTR0,0:CLS1:POKE65496,0:PRINT"Free
```

```
ware Windows BMP Reader 1.1 (C) Septem-
ber 9, 1996 by Stuart Wyss-Gallifent":PRINT
"After loading file, <P> toggle palette reduction
modes, or any other key to quit"
5 INPUT"Source drive (0-3) ENTER=0:";DR:
DR=INT(ABS(DR)):IF DR>3THEN4
6 LINEINPUT"Filename (8 chars.) ENTER=DIR
":F$:IF F$="DIR" OR LEN(F$)=0 THEN
GOSUB 86:GOTO6
7 IF LEN(F$)>8THENF$=LEFT$(F$,8)ELSE IF
LEN(F$)<8THENF$=F$+CHR$(32):GOTO7
8 DSKJ$ DR,17,2,A$,B$:A$=LEFT$(A$,68):
SK=1:F$=F$+"BMP"
9 FOR S=3TO11:DSKJ$ DR,17,S,B$,C$:I=
INSTR(1,B$,F$):J=INSTR(1,C$,F$)
10 IF I=0AND J=0THENNEXTS:GOTO5 ELSE
S=11:NEXTS
11 IF I<>0THEN SK(SK)=ASC(MID$(B$,I+13,
1))ELSE SK(SK)=ASC(MID$(C$,J+13,1))
12 A=ASC(MID$(A$,SK(SK)+1,1)):SK=SK+1:
SK(SK)=A
13 T(SK-1)=INT(SK(SK-1)/2):IF T(SK-1)>16
THEN T(SK-1)=T(SK-1)+1
14 IF SK(SK-1)/2=INT(SK(SK-1)/2)THEN SS
(SK-1)=1 ELSE SS(SK-1)=10
15 IFA<=&H43THEN12 ELSE AA=1:SS=SS(1)
16 POKE&H1806,DR:EXEC&H1600:DSKJ$
DR, T(AA),SS,A$,B$:GOSUB 23:GOSUB51:
POKE65496,0
17 POKE &H1809,T(AA):POKE&H180A,SS:
EXEC &H17DA
18 IF PEEK(&H0E04)=1 THEN HCOLOR J,0:
IF LE<320 THEN HLINE(LE,0)-(LE,191),PSET:
GOTO 79 ELSE 79
19 IF SS<>SS(AA)+8 THEN SS=SS+1:
GOTO17
20 IF AA<>SK THEN AA=AA+1:SS=SS(AA):
GOTO17
21 GOTO 79
22 POKE 65496,0:HSCREEN0:RGB:END
23 ' PROCESS
24 POKE 65497,0
25 PLS=A$: ' SAVE PALETTE DATA FOR
FUTURE RE-COLORING
26 HI=ASC(MID$(A$,23,1)):HI=HI+ASC(MID$
(A$,24,1))*256: LE=ASC(MID$(A$,19,1)):LE=
LE+ASC(MID$(A$,20,1))*256
27 IF HI>192 OR LE>320 THEN PRINT
"ERROR - PICTURE EXCEEDED 320 X 192":
GOTO22
28 IF ASC(MID$(A$,27,1))<>1 THEN PRINT
"ERROR - TOO MANY COLOR PLANES":
GOTO22
29 IF ASC(MID$(A$,29,1))<>4 THEN PRINT
"ERROR - PROGRAM SUPPORTS ONLY 16
COLOR BMPS":GOTO22
30 POKE &H0E03,HI-1:POKE &H0E0C,INT
(LE/256): POKE &H0E0D,LE-(PEEK(&H0E0C)
*256)
31 CLS:P=0
32 FOR H=55 TO 115 STEP 4
33 FOR I=0 TO 3
34 CV=ASC(MID$(PLS,H+1,1))
35 ON I+1 GOTO 42,39,36,45
36 'RED
37 IF CV<R1 THEN CR=0 ELSEIF CV<R2
THEN CR=4 ELSEIF CV<R3 THEN CR=32
ELSE CR=36
38 GOTO 46
39 'GREEN
```

```

40 IF CV<R1 THEN CG=0 ELSE IF CV<R2
THEN CG=2 ELSE IF CV<R3 THEN CG=16
ELSE CG=18
41 GOTO 46
42 'BLUE
43 IF CV<R1 THEN CB=0 ELSE IF CV<R2
THEN CB=1 ELSE IF CV<R3 THEN CB=8
ELSE CB=9
44 GOTO 46
45 'NULL
46 NEXT I
47 PA=(CR OR CG):PA=(PA OR CB)
48 PALETTE P,PA:P=P+1
49 NEXT H
50 RETURN
51 HSCREEN 2
52 J=0:FOR I=15 TO 0 STEP -1:IF PEEK(&H
FB0+i)=64 THEN J=I:NEXT I ELSE NEXT I
53 I=J*16:J=(I OR J):POKE&H177F,J:J=(J
AND 15):EXEC&H177E:RETURN
54 'POKE IN MACHINE CODE
55 IF PEEK(&H161B)=57 AND PEEK(&H17
DA) =8H7F THEN Q1$="FFFF":GOTO 59
56 PMODE 4,1:PCLS:CK=0
57 READ Q1$:Q1=VAL("&H"+Q1$)
58 IF Q1$="FFFF" AND CK<>24968 THEN
PRINT"DATA ERROR":POKE&H161B,0:END
59 IF Q1$="FFFF" THEN R1=64:R2=128:
R3=192: RETURN
60 IF Q1<256 THEN POKE Q2,Q1:Q2=Q2+1:
CK=CK+Q1:GOTO 57
61 Q2=Q1:GOTO 57
62 DATA 1600,8E,11,76,BF,E,6,7F,E,1609,7F,E,
C,7F,E,D,7F,E,4,7F
63 DATA
E,8,7F,E,9,7F,E,3,39,10,21,1620,10,21,1624,
10,21,1628,12,12
64 DATA 7F,FF,D9,BE,E,6,A6,84,B7,E,2,BE,
E,8,BC,E,C,24,3,BD
65 DATA 16,91,BE,E,8,30,2,BF,E,8,7C,E,1648,
B6,E,164E,84,4,81,1652
66 DATA 27,B,7F,E,1657,BE,E,8,BC,E,C,24,
9,B6,E,3,81,FF,27,D
67 DATA 20,14,7F,E,8,7F,E,9,7A,E,3,20,EC,
86,1,B7,E,4,7F,FF
68 DATA D8,39,BE,E,6,30,1,BF,E,6,8C,12,
1687,26,A1,8E,11,168C,BF,E
69 DATA 6,20,E7,B6,E,3,C6,A0,3D,1F,2,FC,E,
8,44,56,FD,E,E,1F
70 DATA
20,F3,E,E,1F,1,8C,20,16AB,24,A,86,30,B7,
E,5,BF,E,A,20
71 DATA
35,8C,40,168A,24,F,86,31,B7,E,5,1F,10,83,20,
16C6,FD,E,A,20
72 DATA 21,8C,60,16CE,24,F,86,32,B7,E,5,1F,
10,83,40,16DA,FD,E,A,20
73 DATA
D,86,33,B7,E,5,1F,10,83,60,16E9,FD,E,
A,B6,FF,A3,B1,E,5
74 DATA
27,6,B6,E,5,B7,FF,A3,FC,E,A,C3,60,1700,1F,
1,B6,E,2,A7
75 DATA 84,39,17DA,7F,FF,D8,BE,C0,6,86,2,
A7,84,B6,18,8,A7,1,B6,18
76 DATA
9,A7,2,B6,18,A,A7,3,10,8E,11,17F7,10,AF,
4,AD,9F,C0,4,7E
77 DATA 16,2A,177E,86,FF,C6,30,F7,FF,A3,
8E,80,00,12,A7,80,8C,80,00,28,F9,5C,C1
78 DATA 34,26,ED,39,FFFF
79 A$=INKEY$:IF A$="" THEN 79
80 IF A$="" THEN 82
81 POKE 65496,0:RGB:END
82 IF R1=0 THEN R1=64:R2=128:R3=192:

```

```

GOSUB31:GOTO79
83 IF R1=64 THEN R1=32:R2=128:R3=224:
GOSUB31:GOTO79
84 IF R1=32 THEN R1=0:CV$="40605074486
8597847645276486D7B7F":FOR H=1 TO 32
STEP2:CV=VAL("&H"+MID$(CV$,H,2)):POKE
&HFFB0+INT(H/2),CV:NEXT H:GOTO79
85 GOTO79
86 POKE65496,0:PRINT:PRINT"BMP files on
drive"DR":PRINT:FOR S=3TO11:DSKJ$
DR,17,S,A$,B$:FORA=1TOLEN(A$)STEP32
87 IFMID$(A$,A+8,3)="BMP"ANDASC(MID$(
A$,A,1))<>0THENPRINT MID$(A$,A,8)
88 NEXTA:FORA=1TOLEN(B$)STEP32
89 IFMID$(B$,A+8,3)="BMP"ANDASC(MID$(
B$,A,1))<>0THENPRINT MID$(B$,A,8)
90 NEXTA:NEXTS:POKE65497,0:PRINT:
RETURN

```

```

1 'PCX READER - BY STUART WYSS-
GALLIFENT (C)1996 JULY VERSION 1.0
2 CLEAR 500,&H5FFF:DIM T(50),SS(50),
SK(25)
3 PMODE4,1:PCLS:GOSUB 38
4 WIDTH80:PALETTE0,0:PALETTE8,63:
ATTR0,0:CLS1
5 PRINT"PCX Reader. (C)1996 by Stuart
Wyss-Gallifent Version 1.0"
6 PRINT:INPUT"Source drive (0-3) ENTER
=0*":DR:DR=INT(ABS(DR)):IF DR>3THEN4
7 HI=0:LE=0
8 LINEINPUT"Filename (8 chars.) ENTER=DIR
":F$:IF F$="DIR" OR LEN(F$)=0 THEN
GOSUB32:GOTO8
9 IF LEN(F$)>8THENF$=LEFT$(F$,8)ELSEIF
LEN(F$)<8THENF$=F$+CHR$(32):GOTO9
10 POKE65496,0:DSKJ$ DR,17,2,A$,B$:
A$=LEFT$(A$,68):SK=1:F$=F$+"PCX"
11 FOR S=3TO11:DSKJ$ DR,17,S,B$,C$
:I=INSTR(1,B$,F$):J=INSTR(1,C$,F$)
12 IF I=0AND J=0THENNEXTS:GOTO6 ELSE
S=11:NEXTS
13 IF I<>0THEN SK(SK)=ASC(MID$(B$,I
+13,1)) ELSE SK(SK)=ASC(MID$(C$,J+13,1))
14 A=ASC(MID$(A$,SK(SK)+1,1)):
SK=SK+1:SK(SK)=A
15 T(SK-1)=INT(SK(SK-1)/2):IF T(SK-1) >16
THEN T(SK-1)=T(SK-1)+1
16 IF SK(SK-1)/2=INT(SK(SK-1)/2)THEN SS
(SK-1)=1 ELSE SS(SK-1)=10
17 IF A<=&H43 THEN14 ELSE AA=1:SS=SS
(1): POKE&H1806,DR:EXEC&H1600: POKE
&HE8C8,141:HSCREEN2
18 FORG=0TO15:READ GG:PALETTE G,GG:
NEXT G
19 DSKJ$ DR,T(AA),SS,A$,B$
20 IF ASC(MID$(A$,4,1))<>8 THEN HSCRE-
ENO: PRINT"ERROR - UNSUPPORTED":
GOTO30
21 IF ASC(MID$(A$,66,1))<>1 THEN HSCRE-
ENO: PRINT"ERROR - UNSUPPORTED":
GOTO30
22 LE=ASC(MID$(A$,9,1))+ASC(MID$(A$,10,
1))*256:LE=LE-(ASC(MID$(A$,5,1))+ASC
(MID$(A$,6,1))*256)
23 HI=ASC(MID$(A$,11,1))+ASC(MID$(A$,
12,1))*256:HI=HI-(ASC(MID$(A$,7,1))+ ASC
(MID$(A$,8,1))*256)
24 IF LE>319 OR HI>255 THEN HSCRE-
ENO:PRINT"ERROR - FILE TO BIG":GOTO30
25 POKE&H0E11,HI:POKE&H0E0F,INT(LE/
256):LE=LE-(PEEK(&H0E0F)*256):

```

```

POKE&H0E10,LE
26 POKE&H1809,T(AA):POKE&H180A,SS:
EXEC&H17DA
27 IF PEEK(&H0E04)=1 THEN 30
28 IF SS<>SS(AA)+8 THEN SS=SS+1:
GOTO26
29 IF AA<>SK THEN AA=AA+1:SS=SS(AA):
GOTO26
30 PALETTE 0,63:PALETTE 0,0:PALETTE0,
63: PALETTE0,0:EXEC44539:PALETTE8,
63:END
31 '
32 'SPECIAL DIR
33 POKE65496,0:PRINT:PRINT"PCX files on
drive"DR":PRINT:FOR S=3TO11:DSKJ$
DR,17,S,A$,B$:FORA=1TOLEN(A$)STEP32
34 IFMID$(A$,A+8,3)="PCX"ANDASC(MID$(
A$,A,1))<>0THENPRINT MID$(A$,A,8)
35 NEXTA:FORA=1TOLEN(B$)STEP32
36 IFMID$(B$,A+8,3)="PCX"ANDASC(MID$(
B$,A,1))<>0 THENPRINT MID$(B$,A,8)
37 NEXTA:NEXTS:POKE65497,0:PRINT:
RETURN
38 PRINT"POKING MACHINE CODE IN..."
39 READ A$:A=VAL("&H"+A$)
40 CS=CS+A
41 IF A$="FFFF" THEN 44
42 IF A>255 THEN PL=A:GOTO 39
43 POKE PL,A:PL=PL+1:GOTO 39
44 IF CS<>190126 THEN PRINT"DATA
ERROR - CHECK DATA STATEMENTS":END
ELSE RETURN
45 DATA 1300,F6,E,A,C1,1,27,10,B6,E,3,48,48,
48,48,B7,E,B,86,1
46 DATA B7,E,A,39,B6,E,3,BA,E,B,7F,E,A,BE,
E,8,A7,84,30,1
47 DATA BF,E,8,8C,80,132D,27,9,B6,E,5,81,
33,27,17,20,30,B6,E,5
48 DATA 4C,20,16,81,33,22,5,B7,FF,A3,20,21,
86,1,B7,E,4,39,8C,78
49 DATA A0,24,F5,20,14,B7,E,5,8E,60,135A,
BF,E,8,20,DF,1368,FC,E,C
50 DATA
C3,136D,2,FD,E,C,10,B3,E,F,23,19,10,83,
1,3F,25,A8,4F,5F
51 DATA
FD,E,C,B6,E,E,4C,B7,E,E,B1,E,11,23,2,20,
B7,39,1600,8E
52 DATA 11,80,BF,E,6,8E,60,1609,BF,E,8,7F,
E,160F,7F,E,C,7F,E,D
53 DATA 7F,E,4,86,30,B7,E,5,7F,E,A,7F,E,E,
B6,E,5,B7,FF,A3
54 DATA 39,7F,FF,D9,BE,E,6,A6,84,F6,E,1635,
C1,1,27,32,81,63,22,22
55 DATA 84,F,B7,E,3,BD,13,1645,7F,E,1648,
BE,E,6,30,1,BF,E,6,8C
56 DATA 12,1653,26,D5,8E,11,1658,BF,E,6,7F,
FF,D8,39,84,3F,B7,E,2,86
57 DATA 1,B7,E,1669,20,DD,84,F,B7,E,3,5F,7F,
E,1,8D,13,1677,F6,E
58 DATA 1,5C,F7,E,1,F1,E,2,26,F1,7F,E,1686,
B6,E,4,81,1,27,2
59 DATA 20,B9,39,17DA,7F,FF,D8,BE,C0,6,86,
2,A7,84,B6,18,8,A7,1,B6
60 DATA 18,9,A7,2,B6,18,A,A7,3,10,8E,11,
17F7,10,AF,4,AD,9F,C0,4
61 DATA 7E,16,2A,FFFF
62 DATA 0,4,2,6,1,5,3,56,7,32,22,54,9,
40,28,63:PALETTE DATA

```



RGBOost - \$15.00

If you want to speed up DECB easily, install an Hitachi 6309 and get RGBOost. This patch for DECB uses the extra 6309 functions for up to a 15% gain in overall speed. It is compatible with all programs tested to date! Save an additional \$5 by purchasing RGBOost along with one of my other products listed below!

EDTASM6309 v2.02 - \$35.00

Patches Tandy's Disk EDTASM to support Hitachi 6309 codes! Supports all CoCo models, including stock 6809 models. CoCo 3 version uses 80 column screen, runs at 2MHz. YOU MUST HAVE A COPY OF DISK EDTASM. This is a PATCH ONLY! It will not work with "disk patched" cartridge EDTASM

CC3FAX - \$35.00

Receive and print weather facsimile maps from shortwave! The US weather service sends them all the time! Requires 512K CoCo3 and shortwave receiver. Instructions for simple cable included.

HSDOS - \$25.00

Move programs and data between DECB and OS-9 disks! Supports RGB-DOS - move files easily between DECB and OS-9 partitions! No modifications to OS-9 modules required.

DECB SmartWatch Drivers - \$20.00

Access your SmartWatch from DECB! Adds function to BASIC (DATES) for accessing date and time. Only \$15.00 with any other purchase!

Robert Gault

832 N. Renaud

Grosse Pointe Woods, MI 48236

313-881-0335

Please add \$4 S&H per order

for all your CoCo hardware needs, connect with

CoNect

1629 South 81st Street
West Allis, WI 53214
(pulland@omninet.uwm.edu)

That thing that Tandy calls a serial port on the CoCo has always been a problem. It was designed with minimal cost in mind, and never upgraded. Even Tandy tried to fix it with their RS-232 Pak, but even it was only half done! Our Fast 232 port uses a 16 byte buffer to alleviate missed characters at any speed and also has ALL RS-232 lines implemented. It is easy to set up with jumpers for different addresses. A daughterboard can be purchased to easily add a second fast serial port! And all this in a cartridge the size of a ROM Pak! 6809 and 6309 OS-9 drivers included. Completely supports up to 57,600 bps, limited support for 115,000 bps.

Fast 232 - \$79.95

Daughter Board - \$45.00

Check with us for complete disk drive systems, misc. hardware items, hardware repairs, and hard to find new and used CoCo software!

Get more from your CoCo OS-9 Level II system with help from...

Chris Dekker

BasicBoost : 6309 port of Basic09's RunB module. Packed programs are 10% to 15% faster (varies with functions used)!

ScreenBoost: 6309 version of Level II screen drivers. Noticeably speeds up most functions! Adds support for up to 200 graphic lines and 28 by 128 column text screens, plus horizontal scrolling. New commands to manipulate graphic fonts and one that allows programs to move and resize the window in which they run.

\$10 each or \$16 for the pair

HSLINK: Null-modem file transfers between a CoCo and any other computer. Uses CoCo printer port -- no special hardware required! ASCII and Xmodem transfers up to 19,200 bps - 57,600 bps with 6309! Cables can be made on request (specify ends needed) or use your own null-modem cables.

\$14.95 or \$24.95 with cable

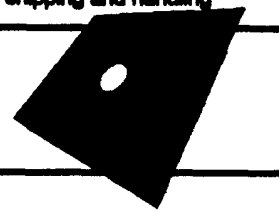
Prices are in US dollars. Call or write for Canadian dollar prices. Add \$3 US, \$5 Canada for shipping and handling

C. Dekker

RR #4

Centreville, NB E0J 1H0

CANADA Phone 506-276-4841



ADVERTISER'S INDEX

<i>Pete Bumgardner</i>	10
<i>BlackHawk Enterprises</i>	13
<i>Chicago CoCoFest</i>	BC
<i>CoNect</i>	21
<i>Chris Dekker</i>	21
<i>FARNA Systems</i>	6,15,BC
<i>Robert Gault</i>	21
<i>Hawksoft</i>	13
<i>Dennis Kitz</i>	4
<i>Small Grafx</i>	13

What are you waiting for?

Get your friends to subscribe to the only magazine that still supports the Tandy Color Computer...
"the world of 68' micros"!

The more people who want the support, the longer it will be here!

GET READY! IT IS ALMOST TIME FOR THE

Sixth Annual "Last"

Chicago CoCoFest

The event of the year is to be held at the Holiday Inn, 345 West River Road, Elgin, IL. If you are just shopping or browsing, come by any time Saturday, April 26 between 10am and 5 pm or Sunday, April 27 from 10 am until 3:30pm. Vendors may come as early as 5:30am on Saturday for booth setup. General admission is \$5 for Glenside members, \$10 all others, one day or both. Vendor booths are \$35.

NOTICE: Due to circumstances beyond their control, the Holiday Inn was forced to give up the rooms reserved for the CoCoFest. They have made arrangements with Red Roof Inn near the fest site (Hwy 59 and I-90... about 10 miles east of the Fest site). A large toy convention in the area precluded closer lodging. Glenside members will be running an informal shuttle service between the two hotels for whoever needs rides. Some of the vendors will also be assisting. Call Red Roof Inn at 847-885-7877 and mention reservation number 1990094096 to get the fest rate of \$60 plus tax. Rooms will be available until April 15th. Glenside apologizes for the inconvenience.

Please send fees to George Schneeweiss, Treasurer, Glenside Color Computer Club, RR#2 Box 67, Forrest, IL 61741-9629. All monies should be received no later than March 6, 1997. Vendors should send \$15 to reserve a booth. Additional booth spaces are \$30, additional vendor passes \$5 (maximum 4 per vendor). Balance of booth rental is due by April 10th, 1997. No refunds after April 10th. For more information contact Tony Podraza, 119 Adobe Circle, Carpentersville, IL 60110-1101. Phone 708-428-3576 or Eddie Kums, phone 708-820-3943, e-mail eddiekums@delphi.com.

Will this really be the "last" Chicago CoCoFest?

Just in case, YOU DON'T WANT TO MISS IT!!!

NEW STUFF FROM FARNA SYSTEMS!!!

FARNA Systems is pleased to announce that we are now distributors of the following, formerly from Northern Exposure! Note: If you never received your order from NX, send a copy of your cancelled check along with \$5 to cover S&H and I'll fill the order! All other orders add \$3 US/Mexico, \$4 Canada, or \$10 Overseas for S&H.

Nitro v2.00 : \$35.00

A complete rewrite of OS-9 Level II that takes advantage of all features of Hitachi 6309 processor. Easy install script! 6309 required.

Nitro Level III : \$20.00

(\$45 if purchased with v2.00)
Just what you always needed... MORE SYSTEM RAM! Up to 32K (16K for RBF, 16K for SCF devices)

TuneUp : \$20.00

If you don't have a 6309, you can still take advantage of some of the Nitro software technology! Many OS-9 Level II modules rewritten for improved speed with the stock 6809!

Rusty : \$20.00

Launch DECB programs from OS-9! Allows loading of some programs from hard drive!
NOTE: DECB Programs will still use floppy drives if required for data.

Thunder OS-9 or

Shanghai OS-9 : \$25.00 each

Transfer your ROM Pack game code to an OS-9 disk! Please send manual or ROM Pack to verify ownership of original program.

FARNA Systems

Post Office Box 321

Warner Robins, GA 31099

Phone: 912-328-7859

E-mail: dertfox@delphi.com