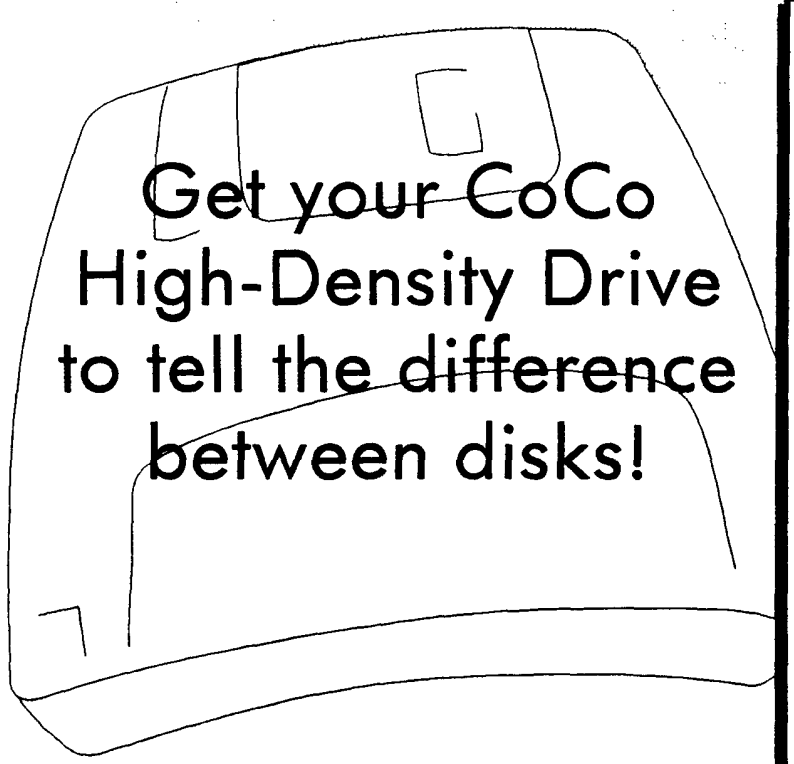
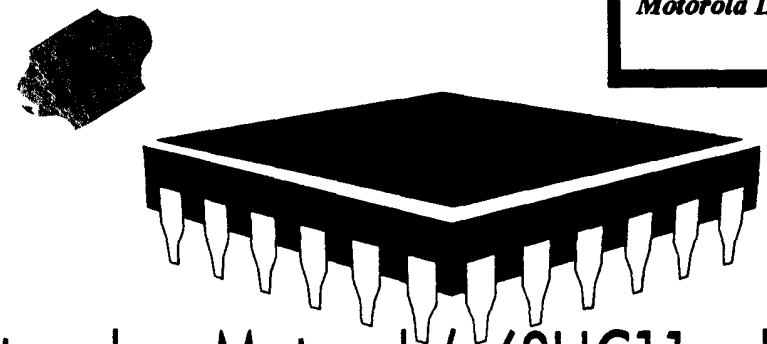
May/June 1996 Vol. 3 Num... $5.00 US

the world of 68' micros

Support for Motorola based computer systems and microcontrollers, a...

CONTENTS





What makes Motorola's 68HC11 a best seller?

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

...ess correction requested

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/ newsstand orders available.

microdisk:
Companion to magazine. Contains program listings, shareware, and freeware as mentioned in magazine text. Three issues per year.
US/Mexico: $20 per year ($8 single)
Canada: $25 per year ($10 single)
Overseas: $40 per year ($15 single)

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

A message from the editor...

I have been trying to get vendors to write about their own products since the inception of "the world of 68' micros". The reasoning behind this is that noone knows the product better than the seller/ developer. And why should anyone else be enthused about a product if the vendor isn't? If you aren't proud enough of your product to tell people you think so, why should they buy? Makes sense, doesn't it?

A lot of vendors just don't seem to agree. They apparently have the idea that if they "toot their own horn" much that potential buyers would think they were being a bit arrogant about it. Well, I think they (I!) should be IF they really do have a good product and it does what it is supposed to do. In all actuality, they don't write because they don't have the time. Here I am, offering 2-3 full pages of free advertising, and they pass it up!

When I first started this magazine, I decided then that it wasn't going to be "my" magazine. It was going to serve the readers and THEIR interests first. The reasoning for this came mainly from on of the AMC auto hobby publications. Although the magazine was supposed to be a club magazine, the editor heavily "flavored" the magazine toward his own views and what he thought the readers should have and think more than what they actually wanted. Up until now, the only place I have expressed my opinion in matters concerning this hobby is in my editorial... simply because that one page is just that, MINE! I have never used more than two pages for my own advertising either.

Two issues bring this up. 1) The fact that an article in this issue is about a new product I am starting to sell, and 2) I recently looked through several years of the old "68 Micro Journal". I've already stated my views on vendors writing about their products. Maybe others will follow this example. As for "68 Micro Journal", I was dismayed that so much content was self serving. I believe that at least 20% of the space was taken up by advertising for the publisher and related companies, and some authors wrote that they were asked to mention items sold by the company in their articles. I'm sorry, but I don't call that being objective or fair to other advert ers. You can count on me not doing anything like that, though I will use a couple pages for advertising.

While I'm on this soap box, let me say some more on where "we", meaning myself and readers, are taking this magazine. For one, we are going to attempt to fill in the gap left in the 68xx/ 68xxx world by the demise of "68 Micro Journal" back in 1990. To be honest, I had that idea way back in '93 when I started "the world of 68' micros". That had a lot to do with my choice of title. Else it could have been "CoCo World" or something with "CoCo" and "OS-9" in it. I did try to limit the magazine to that field of view, but there are no longer enough subscribers to really make the limitation practical. Don't fret, right now I will be publishing for at least two more years, and noone will loose anything. I'll either send refunds or continue until all current subscriptions (at the time) r out and not take any new subs or renew als. I won't simply bailout on you as others have in the past. I didn't like it, and won't do it to you! Don't think you are going to loose any support either. While I'm going to start having some microcontroller and general 68xx and 68xxx articles, as much CoCo support that is there now will continue to be there. An you'll benefit by continuing your much needed support. I have already sent some referals to industrial types needing programmers... referring readers! And there will be more interest by industrial types provided I can prove support for them as well. And they don't mind having the hobby items in the magazine as long as it has something they can use also. So we CAN all get along together.

Please continue your support while we go through some growing changes. As I have said before, this is "our" magazine, and as long as you support it, it will here for you.

Messages from our readers...

Glad to hear your problems are coming around, albeit slowly, and you lookde pretty good at the Chicago CoCoFest. Thanks for the missing page (from "Mastering OS-9"). Renewal money is enclosed for 68' micros and microdisk.

The loss of the thicker covers is minor. I do what you say many others do -- put the magazines in three hole binders. I simply put a stiff divider between each issue now. I like the size and punched holes. You really get the letters scrunched next to each other.

I miss our friend Marty Goodman.

I like your comments on Linux. Now I know why I should stop hoping to be able to run it (or UNIX) on an MM/1B or a lessr cost (and size) machine.

I would like to find a 17 or 18 inch monitor that would do the CoCo 3 and a video tape machine (like the Magnavox 1CM135 does for me now). There was a flurrt of them, but I see no more.

Keep on what you are doing, but more of it.

Good luck,

H. C. Mehlenbacher
Box 430
Grand Marias, MI 49839

H.C., it was good seeing you again in Chicago, as it was the many others who stopped to comment on the magazine or the CoCo and/or OS-9 in general.

I'm not sure about your comment on "scrunching" the letters in... good or bad? I'll take it as good for now, as the tone seems to indicate that you liked the quantity even if it did take a little font maneuvering to get the letters in.

Marty is still in the CoCo world some, but he tends to do a little more with older IBM/PC clones... 8088 and 80286 machines. They are cheap and plentiful now., nobody wants them for much. The CoCo is harder to find, but he still uses them for small projects also.

You can still find monitors in the 17-20" range that will display NTSC composite and RGB signals. They aren't cheap though. Look for anything that has RCA type jacks for video in and will sync down to 15.75KHz. Most of the newer multi-sync monitors won't go that low, but a few designed for video editing will. I don't know of any right off hand, but will keep my eyes open, as I'm sure readers will also.

Please send latest copy of "the world of 68' micros" per your messages on the InterNet. I've been scanning GENie and see you have a disk magazine per old messages on the CoCo RoundTable.

One of my main problems is the lack of a telecommunications program for the CoCo 2. Do you have any available? Or do you know if DeskMatehas Xmodem file transfer capabilities? Also, what all is contained in DeskMate? I am on a budget and can't afford to experiment much with the CoCo as I am already well equipped with Commodore 64/128 equipment and skills. I have contacted Radio Shack but can't get the support I need on current stuff. As a two meter Handi-talkie Ham I can't even get a PS or SWR meter out of Texas or local either.

Ed Cravey
3126 Edgewater Drive
Gainesville, GA 30501

Ed, you'll be getting THIS copy, hope you enjoy it! As you see, this is definately a hard copy magazine. I do, however, produce a companion disk with a lot of the mentioned programs on it. That is published three times per year (see sidebar on page two).

I am also sending you a telecom program. This is Greg-E-Term, and will work on the CoCo2 or 3 well. What you didn't mention is what kind of modem you have. Any external modem will work well with the CoCo serial port up to 2400 baud on a CoCo 3, I don't believe any more than 1200 baud on a CoCo 2. You will need to make a cable for the four pin CoCo serial port as follows (for any CoCo):

CoCo Pin	*RS-232 Pin*
1 (Carrier)	*6, 8, 20**
2 (Receive)	*3*
3 (Ground)	*7*
4 (Transmit)	*2*

**all of these lines are connected to pin 1. Connect all together. If your modem has switches to set the defaults, use pins 4 & 8 instead of 6, 8, & 20. Lines 4, 6, and 20 are tied "high" when connected to pin 8. Switch setup modems can use switches to tie these lines high (4=RTS, 6=DSR, 20=DTR on the RS-232 25 pin connector).*

As for Deskmate, unless you have the manual, forget trying to use it much. The telecom program is especially useless. No Xmodem transfer. You are, however, correct in assuming TEXT is a simple word processor. There is also a PAINT program and a CALC (spreadsheet) program in DeskMate.

If you have a Commodore 64/128 setup, you do indeed have a good deal invested! I personally kind of liked the 128, but the 64 left a lot to be desired. The CoCo was always more versatile, and the disk system much better. And I never did like Commodore's insistence on doing things their own way. More standard peripherals are useable with the CoCo, and that made it a bit cheaper for a hacker to put a good system together. Unfortunately, RS was no better than Commodore in divulging that information... you just have to know whaere to find it! To bad the 128's full potential was never realized.

I won't get into a "flame war" here on the virtues of one machine over another. Each has its advantages and drawbacks. Right now, neither is supported except by users and a few vendors, so both are pretty much in the same boat.

Checks09...

James Jones sends his regrets. Due to a wrist injury, he is limited to the time he can spend typing at the computer. Therefore, the "Checks09" programming project from last issue will be continued in the next issue instead of this one. I do hope that James recovers soon, as I am sure readers do also.

Auto-sense 720K and 1.4M disks...

Mark H. Steiner

Bug Fix and upgrade for the 12 volt floppy controller high density mod!!

This article covers a fix for a problem you may have after doing the high density modification written by Robert Brose in the last issue. My compliments to him for his work on this - well done! After that I will describe an upgrade you can do for 1.4 Meg drives.

THE BUG

The 12 volt controller may develop a problem with the side select signal when the CoCo is running at the 2 MHz clock speed. The side select signal is bit 6 (numbered 0 - 7) of the 8 bit data word used to control a variety of things. The first 6 bits go to a 6 bit latch that works fine. Bits 6 and 7 go to U-10. U-10 is a two flip-flop chip similar to a 7474 except that the pinout is very different. Bit 6 goes to pin 9 of this chip. This is where the problem occurs. At the higher system speed used by OS-9 and some DECB programs this flip-flop will set but will not clear. This causes the controller to stay on the top side of the floppy disk and can't get to the start of the root directory on the disk. I do not know why this happens because this part of the controller is not changed by the modification.

THE FIX

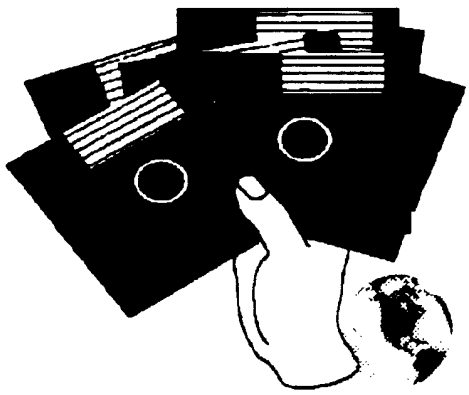
To fix the side select problem I decided to use another 6 bit latch since that seemed to work well. I used a 74LS174 for this. To install the chip bend up all pins except 1, 8, 9, and 16. You can clip all the bent up pins except 14 and 15 if you want. Place this chip on top of u-8. Be sure to align pin 1 of this chip with pin 1 of U-9. Solder pins 1, 8, 9, and 16 to U-9. Next we have to route the side select signal to the new chip. Follow the trace from U-10, Pin 9. It goes to a solder pad near pin 1 of the main 40 pin 1793 controller. It is the closest solder pad to the solder joint between resistors R10 and R12 . Using a small wire (wire wrap wire if possible) trim a small amount of insulation from one end. Heat this pad with a soldering iron and put the striped end of the wire through this pad. It is a hollow sleeve that goes through the board. Trim of any extra wire that sticks out below the board. Solder the other end of this wire to pin 14 of the new 74LS174 you installed earlier. Then solder a wire to pin 15 of the same chip. Carefully remove U-3 from its socket and gently bend up pin 11. Place this chip back in its socket. Be sure to watch for pins that bend when you put U-3 back in. Solder the other end of the wire from the new 74LS174, pin 15 to U-3, pin 11. Check your work once again to be sure it is correct. These controllers are hard to find and you don't want to damage it. If everything looks good then you controller should be fixed. Format a disk and copy some large files to it.

THE UPGRADE

While I was very pleased with the original modification I was a bit unhappy that the 1.4 Meg disk could not be used for 720K disks even though the drive itself could was capable of doing so. After a bit of testing I found a way to fix this (NOTE: this upgrade works only for 3.5 inch drives and only for one drive on the same ribbon cable). Underneath the drive on the right side is a switch that tells the drive if the disk in it is a 1.4 Meg or 720K disk. Use a volt meter to see which side of this switch changes voltage when you insert a high density disk. My drive goes to 0 volts when a high density disk is installed and 5 volts with a 720K disk. If your drive is opposite from this the patch described below will not work. You will have to combine the disk type and drive select signal with a two input AND gate chip and wire it as required. The next step will depend on the drive you have. You need to find an unused pin on the drive's data connector and run a wire from the side of the switch that changes to this pin. This pin must also not be used by the floppy controller. I used pin 34. This is the d' ' changed signal that tells some cont lers that the disk has been replaced with another disk. Our controller does not use this option. Whatever pin you use, to be safe you need to remove the same pin from all other connectors on the same cable so that some other drive does not interfere with this signal. Next you will need a 7400 chip. Bend up all pins except 7 and 14. Solder pins 9 and 10 together. Next solder pins 12 and 13 together. Now place this chip on top of U-6. Make sure it is pointed the right way. Solder pins 7 and 14 to U-6. Run a wire from the pin 9&10 connection to whichever pin you used to get the disk type signal back up the drive data cable. In the normal, if you used the drive select method, you ran a wire from the chip on top of U-7, pin 1, to the drive select pin desired on U-2. Remove the end of the wire from pin 1 of the top chip on U-7 and solder it to pin 4 of the upper chip on U-6. Solde short wire between pins 5 and 8 of the same chip. Then solder a wire from pin 6 to the pin 12&13 pins. Finally run a wire from pin 11 of the upper chip on U-6 to pin 1 of the top chip on U-7.

Your drive should now read a 720K or 1.4Meg disk when you put it in. Enjoy working with this upgrade. It is a nice addition to a fine modification. For questions I can be reached via E-mail as MODEL299@Delphi.com or by U.S. mail at the following address:

Mark H. Steiner
3101 Washington St #74
Bellevue NE 68005



The AT306 Computer

Frank Swygert

OS-9/68000 doesn't come any cheaper, or possibly better, than this!

Karl Kreider has had his AT306 motherboard and accompanying version of OS-9/68000 for about a year now. When he was developing this board, he asked me what I thought of his ideas, being as I heard a lot from readers. Some of those ideas had an impact on the final design, I am pleased to note. Karl also asked me about becoming a dealer. I had some reservations on this at the time. There were to be at least two people selling them already, Wittman Computer Products and BlackHawk Enterprises, both now advertisers, and BlackHawk had a west coast dealer. I thought that this may be enough to meet the needs of most people. Another problem was the amount of support I could provide. I was already juggling a magazine, a software/hardware company, a job, and a family. And I'm not that fluent in OS-9 either. Oh, I can get around in it, but not good enough to support a new system that may have some bugs crop up.

Well that has changed now. The system has matured enough that I can assemble a reliable product from readily available components. The result is in the FARNA Systems ad in this issue. Sure, many of you could buy the components separately and assemble the same system, with a savings of around $150. But you may have a few problems getting everything to work, or you'll have to wait for this or that to come in. Purchasing from me, you get a complete, plug-n-play system with full documentation and a 90 day warranty (more on some components). The folowing information applies to all the AT306 based systems, not just those from FARNA.

Just what is the AT306 motherboard? Well, to begin with it is a small footprint (about eight inches square) board with holes drilled to fit a standard PC Clone type case. It will fit in almost ANY case designed to hold a "standard" motherboard. Only some of the Tandy, Packard Bell, and other custom made cases won't hold one of these boards with no modification.

Built on this motherboard is all the basic input/output functions needed. There are three to five 16 bit ISA bus connectors (standard "AT" types) for additional functions. The basic functions include:

* IDE hard drive interface for two drives
* High density floppy drive interface for two drives
* four 30 pin SIMM sockets (linked in pairs, 512K to 16MB supported)
* two high speed (up to 115K baud) serial ports
* one bidirectional parallel printer port
* real time clock
* standard "AT" (5 pin) keyboard connector

The only function NOT built into the motherboard is the VGA video adapter. When starting the AT306 project, Karl had thought that a single driver would support at least the basic functions of almost any VGA card. Unfortunately, this did not turn out to be the case. Each video card tested varies a good deal from the "standards". The clones, being hardware mapped to drive such things, seem to have a tolerance for these variations. OS-9, with drivers written for each device, is more precise. This means that drivers have to be modified for each make and model of video card. Currently, the Trident 8900CL and the Hercules Dynamite Pro are supported. The Trident is a very generic, mid range performing graphics accelerator type card. The Hercules is one of the faster accelerators. They are, of course, priced accordingly. The Trident retails for around $60, the Hercules about $100 more.

Many of you with PC experience will note that many manufacturers have special drivers written for their cards. These drivers take advantage of special features built into the chipsets they use. The cards will display basic VGA even without the special drivers though. On this note, I'll add that some other high end cards should work with the AT306. The Hercules card uses a Tseng 4000 W32i chipset. According to Karl, any card with this set should work with the supplied drivers. While researching prices myself, I have found that the current most common Tseng 4000 chipset is the W32p. DO NOT buy one of these, as they may not work. I'll be contacting Karl about this later, but the W32p chipset may be for VLB or PCI cards only.

Having a standard 16 bit ISA expansion bus also opens the machine up for other peripherals. Currently, several SCSI cards are supported. Simply add a Future Domain 1830 or easy to find Adaptec 15xx controller and open yourself up to a wide range of additional peripheral devices. SCSI CD-ROMs and tape backup drives (150-325MB Wangtec, TEAC, Archive, or Connor with DC600 type cartridges) are known to work with the provided drivers. Sound Blaster sound cards (record and play sound in DSP mode) are also supported. The Sound Blaster SCSI card uses the NCR5830 SCSI controller, which can be used with CDs and tape backup drives.

The AT306 gets its name from the processor it uses — Motorola's 68306 microcontroller. A microcontroller was used because it has a lot of the necessary computer components built directly in the chip, such as ROM and RAM control. This further reduces the chip count, number of solder joints, and total size and complexity of the motherboard. The 68306 is basically a 68000 chip with the added components. It runs at a clock speed of 16.67MHz. A 25MHz version is available, but Karl says it can't just be plugged into the motherboard, and changing the crystal would alter everything else also.

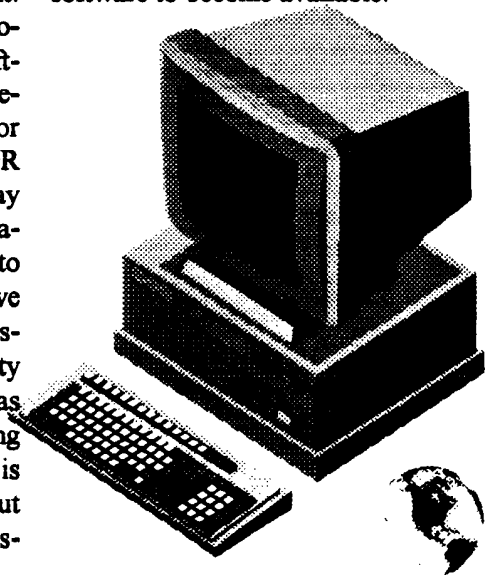
Karl was thinking about future expansion when he designed this board. There is an upgrade processor slot designed to take a small daughterboard. There is a possibility that a 25MHz 68306 board will be released, but the most liekly future upgrade will use a 68340. This microcontroller uses a more powerful 68020-like core. I say "like" because there are a few functions of the 68020 not implemented in the controller processor core. This means that some, but not many, programs written for a 68020 won't run. The majority, however, will, as the missing functions aren't used much. Hopefully, Karl will make an upgrade available that uses a memory management unit at a later date. This will make Linux portable to the board, even if it doesn't support additional memory.

The OS-9 version sent with the machine is "Personal" version 3.0. I put personal in quotes because such a thing doesn't officially exist. A license for the normal version of OS-9 sent with industrial computers was just to much for the average hobbyist. Peripheral Technology sells OS-9 for the PT68K4 68000 based machine for $300... that's just the operating system, the motherboard is just as much! Karl wanted to keep the price of the motherboard and operating system down to no more than $400 to the dealers. To do this, he purchased a license for the Industrial version of OS-9/68K from Microware. The industrial version is intended for embedded, ROM based designs with limited input/output capabilities. He then added individual licenses for a few of the necessary file managers and drivers, and proceeded to write the remianing drivers and managers himself. This resulted in a very compatible, affordable version of OS-9/68K. It will run 99% of the generic software and utilities that are readily available for downloading from various sources, including Delphi and the Internet (a good ftp site is sysconintl.com).

Software that comes with the system includes a version of BASIC that is very similar to Microware BASIC (and Basic09) and a graphical user interface (GUI) called "MGR" (ManaGeR). Karl ported the 1988 freeware version to the AT306 to give the user a very affordable graphics based system. MGR works much like MultiVue on the CoCo and Windows on a PC. It has a look and feel similar to both systems, and is very easy to work with. There is a commercial version available in Europe. I have contacted some people in Germany about the possibility of getting some MGR software. And I have also asked about the compatibility between the freeware and commercial versions. There is one minor problem with the shipped version of MGR. If you switch to a terminal window while you have multiple MGR windows open, the MGR windows "disappear" when you switch back! They don't close or go away, the screen just doesn't update them when you switch back. If you remember WHERE the windows were, you can "click" on them and they will update. I personally see this as a minor problem. Most of us either work in terminal windows or the GUI, not both. Yes, I know some of the more advanced users will say "not so!", but the "regular" fellow won't have a problem with this. And as long as the problem is known and pointed out, it can be easily worked around.

What I have done is put together a complete system that I can easily support and ship out as a "canned" unit. Basically, this means that all components are assembled and tested, software installed and running in a "generic", ready to use configuration (4 or more terminal windows and one MGR window). A general purpose user may never have need to alter this configuration. And the system is guaranteed to operate as configured when you receive it. The only restriction is that the system not be altered for the first thirty days. I can only guarantee it to work as I set it up. Once a user starts making major changes to the system, there is no way I can back-track and find out what has been altered to cause the system to crash. All I can do in this case is recommend that you restore the backups made when you received the system. You should ALWAYS backup a "canned" system when you receive even though system and software disks are included. This assures that all configuration files are saved as they were, and the system can be easily restored to its original configuration. When you do start making changes, make them one at a time, but backup the files you intend to alter first so they can be restored if the changes don't work out. This is time consuming, but you'll easily be able to discover what does and doesn't work, and even restore the system easily.

FARNA Systems prefers payment with an order for hardware. Please allow a minimum of 30 days, maximum of 45 for orders to arrive. We sometimes have delays getting parts, but should be able to easily deliver in this time frame (or sooner). We cannot accept credit card orders at this time, but personal checks drawn on a US bank are welcome. I'll add one more thing here. While FARNA does not current sell individual components, the other two dealers in the magazine do. If you think our terms to restrictive, please contact them! We currently have a good working relationship with the BlackHawk and Wittman, and intend to keep things that way! See out ad for pricing, and watch later for additional software to become available.



The Fifth Annual 'LAST' Chicago CoCoFest

No, it's still NOT the last one!

Frank & Tiffany Swygert

Frank:

The very best thing I can say about the last Chicago CoCoFest is that it isn't *the last* one! Even as we got ready to leave, the date was announced for next year... April fourth and fifth, 1997.

Well, what about this past fest? There were 19 vendors who made an appearance this year. That's about as many as the 1995 fest. As long as we can continue to draw this many, there will continue to be CoCoFests in Chicago.

As always, this year's fest was held at the Elgin Holiday Inn, which is a Holidome Recreation Center. There is a large whirlpool, a heated indoor pool, an excercise room, and a small arcade. Plenty of indoor room to keep the kids busy while the adults peruse the show or get somr rest and relaxation. This is one of the things that make the Chicago fest a great family trip. And there are plenty of good sites and shopping areas nearby without going into the big city.

What most of you want to hear about is all the new items. Well, there weren't a whole lot this year. Digigrade productions was showing off a new adventure game, DAS, for the MM/1. It is similar in concept to "Doom" on the IBM/PC and clones. Bill Wittman was showing off the AT306 and an accompanying POS (Point of Sale... replace tha cash register!) system. He also had some new software. Strongware also had a few new games for the MM/1 available, and will be selling Sub-Etha products in the future.

Sub-Etha was sorely missed this year! As a company, they will no longer be with us. As mentioned before, however, Strongware will become a distributor. I spoke with Allen Huffman at length about Sub-Etha this year. Basically, Allen though that running a software company while working for Microware could be a conflict of interests. That and the fact that Microware keeps him running all over the country leaves him little time to program, much less keep up with incoming orders and support!

Did I mention that Allen now works for Microware? He found out that Microware was looking for someone to train people on using OS-9 from another Microware employee, who urged him to submit a resume. No harm in that, so he sent one in. A few weeks later he received a phone call and found himself in the middle of an impromptu interview. This led to Microware sending him a plane ticket from Texas to DesMoines for a personal interview. He thought it was a little touch and go with one interviewer, but another was apparently impressed. Needless to say, a few weeks later he was on the move to DesMoines, Iowa, and a new job as a trainer with Microware. Congratulations and good luck, Allen! He will continue to write some programs and market his existing work through Strongware and is open to marketing through other dealers.

As always everyone had a good time meeting old friends and making new aquaintences. The fest has very much become a social gathering among vendors, but that doesn't mean that it isn't a profitable show!

Personally, FARNA Systems brought home a couple hundred after paying all expenses. This isn't bad, as it costs about $600 for me to attend. I had expected to gross from half to two-thirds that amount., so I was very pleased.

My feeling towards the fest in Chicago is that as long as I can meet at least half of my expenses, I'll continue to attend. For one thing, I owe it to you, my readers. By continuing support, you make it possible for me to attend the fests, and I believe I should be available whenever possible.

There was some discussion about whether to have an Atlanta fest or not, or to maybe have a fest about the same time but another location, like maybe Pennsylvania or the Princeton, New Jersey area (where many of the old RainbowFests were held). These are viable locations, as there are a good many northeasters who would probably attend and can't make it to Chicago due to the long distance. When asked if I would attend a fest in such a location, I had to be very skeptical. Chicago has become THE event of the year as far as the CoCo and OS-9 are concerned. While I feel I have an obligation to attend Chicago, I can't afford two long distance trips every year. If Atlanta weren't literally in my back yard, I would have difficulty attending it every year. I could probably attend a fest no more than a six to eight hour drive away, but any more would be extremely difficult.

I've noticed that some of you do bring wives and/or girlfriends with you, but many don't. Well, this year Tiffany, my bride of only five months at the time of the fest, came with me to Chicago. She spent a lot of time with me helping with the booth and allowing me to look around a good bit, but she also went off with Linda Podraza a few times to do some shopping and sight seeing. I'll let her tell you a bit about that, and the Saturday night social gathering of vendors.

For those of you who remember the Rambler from last year, it is still with me. It didn't make it to Chicago this year, mainly due to a bad axle bearing which I'll get fixed this summer. We didn't have to take the truck (mentioned in the last issue's editorial) either, because we bought a little Subaru station wagon just weeks before the fest.

This year the vendors got together on Staurday night instead of the traditional Sunday-after-the-Fest dinner. Thanks to Paul Jerkatis (Sand V BBS) for organizing this! We had been going to a Mongolian Bar-B-Que, but Paul had a surprise for everyone this year! We all went to the "Prairie Rock" brew-pub in downtown Elgin. It was much closer and a very nice adult club. They make their own ale, and you can see the micro-brewery in action behind glass walls. The ale was excellent, the food great, and the atmosphere very pleasing. Thanks for making this year something special Paul!

I'll close with a list of attendees and a message: Don't forget to attend if you can next year! Without continued support, the fests will cease. We need you all!

(Vendor list on page xx)

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
Update of Paul Ward's "Start OS-9". Completely steps one through learning all aspects of OS-9 on the Color Computer. Easy to follow instructions and tutorials. Comes with a disk full of added utilities and software!

Tandy's Little Wonder - $25.00
History, technical information, hacks, schematics, repairs... almost EVERYTHING available for the Color Computer all under one cover! A MUST HAVE for those wanting to keep their CoCo alive or for those new to the CoCo.

Quick Reference Guides
These handy little books contain the most referenced information in an easy to find format. Small size makes them unobtrusive on your desk. Command syntax, error codes, system calls, etc.
CoCo OS-9 Level II - $5.00
OS-9/68000 - $7.00

SOFTWARE:

CoCo Family Recorder: The very best in genealogy record keeping EVER for the CoCo! Requires CoCo3, two disk drives (40 track for OS-9) and an 80 column monitor.
DECB: $15.00 OS-9: $20.00

DigiTech Pro: $10.00
Easily 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: The premiere, most respected enhancement for DECB! Support for double sided drives, 40/80 tracks, faster formatting, 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 programmmers!

Patch OS-9 - $7.00
The best enhancement for OS-9 short of buying a 6309 and NitrOS9! Contains the latest version of all popular utilities and new commands with complete documentation. Has auto-installer (requires 2 40T DS drives, one may be larger). Ask for 35 track disks (manual install).

HARDWARE:

SPECIAL PRICE ON REMAINING VIDEO DIGITIZERS: $100 (was $125)
This may be your LAST CHANCE to get a DigiScan video digitizer! No MPI -- uses joystick ports. CoCo Max3 , Max 10, Color Max 3 compatible. Input from any VCR, camcorder, or TV camera (freeze-frame required).

Special Prices on Remaining DISTO stock! LIMITED QUANTITIES! BUY NOW!
Mini Controllers: Clones of the Tandy short disk controller, comes with DECB 2.1. NO CASE. $40 each
512K Upgrades: With RAM chips: $50, without: $25
Super Controller I : I have several boards that need repair. Buy two for $30 so you'll have extra parts, or one for $20. Comes with schematic. NO WARRANTY.
HD Controller: $30 - Connect embedded SCSI or MFM/RLL drives (with Adaptec controller) to your CoCo. These fit inside a Super Controller I or II.
Bare Board Set: $30 - Set of blank circuit boards for Disto products including SCI, SCII, HD, 4-n-1, MPROM, etc. Add $20 for complete schematic set.
Complete Disto Schematic set: $25.

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 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
BASIC (resembles Microsoft BASIC)
MGR Graphical Windowing Environment 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

Many other utilities and tools

FARNA-11122 Includes:
2MB RAM
200MB Hard Drive*
Trident 8900 1MB Video Card
$960.75

FARNA-11225 Includes:
2MB RAM
500MB Hard Drive*
Tseng W32i 1MB Video Card
$1114.47

****This is the SMALLEST amount of formatted space (most manufacturers specify UNFORMATTED CAPACITY). Prices fluctuate - we get you the largest drive possible for the money allotted!***

Call for a quote on different configurations and components. Warranty is 90 days for labor and setup, components are 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**
More software will be listed later!

Operating System Nine

Rick Ulland

Why an operating system at all?

Magazine columns share a basic problem. The writer has read them all. Not wanting to alienate long time readers with repetition, the content tends to degenerate into how-to procedures for some obscure minutiae that the average user never even heard about, much less needs to adjust. Before we drift any further thataway, it's time to step back and look over the basic problem again, starting with, 'I need to run these programs'. What's it going to take? At the least, all programs should be able to understand the same disk format, and simple tasks like reading the keyboard shouldn't be left to the individual. Beyond that, there are many ideas of what an operating system should be.

Most can be divided into two groups. The simplest to understand are the bare bones, let's make the programs job a little easier systems like CP/M, MSDOS, or a host of 'personal computer' systems like DECB. Here, the application program reigns supreme, with total control over the machine and all it's resources. They are single user/ single tasking almost by definition, since the 'operating system' is little more than a library of subroutines that programs call upon to perform basic tasks like saving a file on a disk other programs have used.

Such a system has advantages. The only thing going on is the primary task, so there is no overhead. The opsys only exists when app code jumps into it for some standard service. Such a system _can_ be small and fast, and doesn't have to be very complete. As long as the opsys covers the majority of commonly used functions, any special needs an application requires can be provided by the application itself. Or by added hardware- PC clone add-ins commonly contain ROMs to insert additional code to MSDOS. As long as these extra drivers exit with the machine in the same approximate state as when they were called, compatibility with most other software is likely.

This simplicity evaporates when a second user or process is added. With any program compatible of leaving the machine in any state, it becomes very difficult to switch between applications on the fly. Even when there is a protocol for doing so, the unrelated programs might not be able to guess what was going on when they were called up- witness the interesting things that can happen after adding a few badly matched TSRs ('terminate and stay resident programs') to an MSDOS system. And of course, it's impossible to do more than one thing at the same time without extensively rewriting each application to support task switching.

OS-9 is supposed to deliver true multitasking, with no interaction between separate applications. Even when two users write to the same disk at the same time, or when data is coming in the serial port while the terminal program is not in it's timeslice. Obviously, the opsys has to have complete control of the machine. App code can't just jump off to some canned subroutine whenever it wants to write a disk sector- there has to be a master which the applications cannot ever override.

Which means an application shouldn't have access to the physical machine at all. To allow this, system services are requested by issuing a 'syscall', and OS-9 itself can then take care of the actual operation. One major advantage is the application doesn't have to worry about what other apps are doing, how often it's getting a timeslice, what's happening to the hardware when it's 'sliced out'.... But this utility comes at a price. All the extra work saps system resources (something a 2Mhz machine doesn't have that much of) and controlling the system can be something of an art. Really, OS-9 isn't hard on purpose!

With a number of applications all waiting for service, data coming in from the hardware, and the need to run the opsys code itself, there obviously has to be some sort of scheduling scheme. After all, computers still tend to do one thing at a time. By far the easiest part of this problem is making sure every program gets it's fair share of processor time. You or the programmer has the privilege of assigning each process a priority, or starting age. As each one gets a timeslice, the others all age a little bit, and the one that just ran is reset to it's priority. The next slice is then given to the oldest process that's active. You'll note this isn't exactly a line, since process aren't ran in any set order, but all compared anew at the start of each slice. If a process has it's priority set high enough, it may run dozens of times before any lower level process becomes old enough to get a shot.

As clever as this scheme is, its not enough. No matter how often a given process gets CPU time, there is a good chance it's going to be dormant when something important happens- or it's going to waste cpu time when nothing is going on. To avoid spending time on programs with nothing to do, they will assume three states- either active, waiting for something to happen, or sleeping for a definite time (unless something happens first). Which means there has to be a way for process to communicate with each other, the hardware, and the opsys itself.

First up is the process to process signal. A signal is simply a numeric code, which will be stored in the descriptor of the receiving process. If the receiver wasn't active, it's woken up, and when it gets it's next timeslice execution jumps to that process' internal signal handler- which can then read the signal's value and act appropriately. The obvious problem is nothing happens when the signal is sent- it's not until the *receiving* process gets a timeslice that anything can be done. To avoid this limitation, there are also software interrupt. Each process can set up three separate restart points (although OS-9 syscalls use one of them). When a matching IRQ is sent, OS-9 drops everything and immediately jumps to the appropriate location.

Hardware can signal to a process also, through the hardware interrupt. Like the software irq, these are immediately serviced but unlike soft IRQs, there is only one and it's shared by the entire system. To make sense of it all, the opsys maintains a table of devices that might send an irq, and 'polls' each device in turn to see which sent the current one. To reduce the amount of polling needed, device drivers can set a priority, ensuring those issuing many irq or requiring very fast response are placed at the head of the line.

Polling works fairly well- consider the basic serial port, which only buffers one character. It has to send an IRQ and wait for the poll, then some service from the machine, then do something with the character that eventually appears. Finally, another letter can be caught!. Stock OS-9 on a stock CoCo can do this about 600 times per second.

As with any operating system, the user needs a way to make the machine do things when an application program isn't running. Especially in a multiuser/task scheme, where the master has to be insulated from the hoi palloi (users), this means an added step. Under OS-9, it's a program called 'shell', which provides the familiar OS-9 prompt and patiently waits for your instruction, which is executed using the normal syscall requests. Changing the shell can drastically change the look of an operating system- consider the stock version vs. MultiVue's gshell. Shell also provides the most basic example of a process in wait mode- it will start another program, requesting this 'child' signal it when the child is terminating, then it goes into hibernation, using no cpu time but instantly available once the child process ends.

This is quite a bit to keep on tap. Even though the CoCo3 can have up to 2 megs of RAM, the cpu still enjoys a 64K limitation. The easiest way to overcome this is to split the total of things to do into 64K sections- these sections *usually* form a complete 'program'. The operating system itself is one such program. Each additional task running occupies one of these not-to-exceed-64K spaces, and this is a real limitation nowadays. There is another form of interprocess communication available to work around this, the pipe. Pipes simply connect the output of one process to the input of another. Think of pipes as a way to have a program do the keyboard entry for a second program, and you are very close.

Multiple, completely independent 64K programs all running at the same time sounds pretty good- but this still puts a limit on small machines, especially those with more than one user. If a large application is used by more than one person or window, precious memory will be wasted as the exact same code is loaded into multiple locations. Enter the module- a small (usually), independent section of code that performs some specific task. By storing all data outside the program modules, users can share the same copy,and the amount of memory saved can be considerable. The CoCo's MMU makes this 'reentrancy' easy to accomplish. Each 64K space consists of eight 8K blocks, which are stored as a 'task register'. Some of the blocks are used for data, and these are usually unique to a specific task, but any program block can be mapped into several tasks by simply placing the same physical block address into more than one of these task maps. If you then add up all the blocks being used by all the separate tasks, the system may appear to be using more ram than it actually has!

This does leave one problem- the blocks are a fixed size, and it's pretty large. Programs would like to see hundreds of tiny modules they could pick from, instead each task has only 8 storage blocks, which have to be divided between code and data. Microware left a way around the problem, in that an entire disk file is loaded together, into as few blocks as possible. This group gets mapped around as a unit. If it is smaller than 8K, the extra modules are 'free' since 8K would be used in any case. Once a merged file gets larger than this minimum size, things get tougher- Even one byte over, and an extra block has to be added! The extra baggage will mean less data space in any given task map, and the art is in guessing which modules will be used together along with the data space that they will need- hopefully leaving some room for additional modules to be added in when a more complex task is built up.

Not quite impossible, if you know what's needed! Let's use a simple example- assume all the disk utilities except dsave (which is almost 8K itself) are merged into one file. This comes to three blocks, so it looks like there are five left. But keep in mind, you always have shell, which will also need at least one data block- five. Now add an application of only a few blocks size, and copy doesn't have any extra data room- it dragged all that spare baggage along, filling up the 64K task map with unused code. The key, should you choose to accept it, is to put copy (or other dataspace hungry modules) in a separate file that doesn't run over 8K. This way, anything that maps in one of these heavy hitters is only penalized the default one block, and has the bene[f] of the maximum number left free in it task.

You'll probably note we are playing *very* fast and loose with memory requirements here- the cpu can only handle 64K, and we've given the whole amount to the application and it's data- now add the operating system- and the screen has to be stored someplace, doesn't it? Well, yes. The opsys is easy- it's a separate process, so your 64K program has the full use of a 64K operating system- passing syscall requests through a common area at the top of the last block of each task map in the machine. This common area is why merged program files should be a page smaller than a full 8K DAT block... But what about graphics? Obviously, there is too much work involved in a graphics screen to pass through a small data window- and a 32K screen is way to large to fit inside the operating systems space.

More clever work. The grfdrv you see in a mdir isn't really doing anything. It's there to be copied- into a screen ma Like any other module, it's not really copied, but reentered in yet another task map. Graphics functions are issued as syscalls, which OS-9 passes to the copy of the graphics driver that lives with the screen involved. Once this map becomes active, the disjointed grfdrv can quickly modify the large number of pixels involved- we've reduced the whole into a large gfx screen and equally large driver, with no application and one simple task to perform- just barely small enough for a 6x09 to handle. This leaves an application a few steps removed from it's display and perhaps a little slow, but all the above allows a single application to use 196K of ram under a 64K cpu- with the reentrancy, five or six applications can easily use 196K each under 512K of total RAM.

Next month, we begin to explo[re] how this is used....

Rick can be reached for comment here at the magazine or through electronic mail at:
pulland@omnifest.uwm.edu

Roadmap to the Internet

Patrick D. Crispen

Internet Security

"Cyberspace, in its present condition, has a lot in common with the 19th Century West. It is vast, unmapped, culturally and legally ambiguous, verbally terse (unless you happen to be a court stenographer), hard to get around in, and up for grabs. Large institutions already claim to own the place, but most of the actual natives are solitary and independent, sometimes to the point of sociopathy. It is, of course, a perfect breeding ground for both outlaws and new ideas about liberty." —

John Perry Barlow,
Crime and Puzzlement

I would love to tell you that the Internet is a safe place and that there is no reason for you to protect your password. Unfortunately, there are a LOT of people out there who would LOVE to break into your account and "use your account as a base for operations." How prevalent is this? According to Mike Godwin, Chief Legal Counsel for the Electronic Frontier Foundation, it's "fairly common." (1)

The main defense against people who want to break into your account — a.k.a. "crackers" — is your password. Keep your password secure, and you should never have anything to worry about. Give your password to others, or write your password down and put it near your computer, and ... well, you get the picture.

There are some KEY points you need to remember to protect yourself and your account:

- NEVER give your password to *ANYONE*. The whole purpose of having a password in the first place is to ensure that *NO ONE* other than you can use your account.
- NEVER write your password down, and especially never write your password anywhere near your computer.
- NEVER let anyone look over your shoulder while you enter your password. "Shoulder Surfing" is the most common way that accounts are hacked.
- NEVER e-mail your password to anyone.
- DO change your password on a regular basis. There is no better way to thwart a would-be cracker than to change your password as often as possible. Your local Internet service provider will be able to tell you your system's recommendation on how often you should change your password, but a good rule of thumb is to change it at least every three months.
- DON'T pick a password that is found in the dictionary. When you set your password, it is encrypted and stored into a file. It is really easy for a "cracker" to find your password by encrypting every word in the dictionary, and then looking for a match between the words in his encrypted dictionary and your encrypted password. If he finds a match, he has your password and can start using your account at will.
- DON'T use passwords that are foreign words. The hacker can get a foreign dictionary, and ...
- NEVER use your userid as your password. This is the easiest password to crack.
- DON'T choose a password that relates to you personally or that can easily be tied to you. Some good examples of BAD passwords are: your name, your relatives' names, nicknames, birthdates, license plate numbers, social security numbers (US), work ID numbers, and telephone numbers.
- DO use a password that is at least eight characters long and that has a mix of letters and numbers. The minimum length of a password should be four to six characters long.
- NEVER use the same password on other systems or accounts.
- ALWAYS be especially careful when you telnet or rlogin to access another computer over the Net. When you telnet or rlogin, your system sends your password in plain text over the Net. Some crackers have planted programs on Internet gateways for the purpose of finding and stealing these passwords. If you have to telnet frequently, change your password just as frequently. If you only telnet occasionally, say, for business trips, set up a new password (or even a new account) just for the trip. When you return, change that password (or close out that account).

The best passwords — the ones that are the easiest for you to remember, and the ones that are the hardest for crackers to crack — are passwords that are like those fake words you used to create when you would cram for a test. For example, to remember that "the Law of Demand is the inverse relationship between price and quantity demanded," I created the word TLODITIRBP&QD. NO ONE could hack that as a password. Best of all, its EASY to remember (well, its easy for an economist to remember). Here are a couple of other good passwords:

Sentence	Possible password
In 1976 I moved to Tulsa, Oklahoma	I76IMTTO
The conference lost 12,000 dollars	TCL12KD
U of A Crimson Tide Football is #1	UACTFI#1

Sentences are EASY to remember, and they make passwords that are nearly impossible to break (and please do NOT use these sample passwords as your own). Do NOT use well known abbreviations (for example: wysiwyg), and do NOT use keyboard patterns (for example: qwerty) as your password.

If you notice weird things happening with your account:

1. Change your password IMMEDIATELY!
2. Tell your local Internet service provider about it.

It is very common for someone whose account has been hacked to dismiss the signs that the account has been hacked as technical problems with the system. However, when one account is hacked, it very often puts the whole system at risk.

Finally, there is one last thing that I want to say before I close: I feel that "hacking" and "cracking" so violates the spirit of the Internet that I will do everything in my power to help put the overgrown babies who engage in such activities where they belong — behind bars. Until that time comes, however, I'm going to change my password as often as possible.

Homework

Contact your local Internet service provider, find out how you can change your password, and CHANGE YOUR PASSWORD!!

Patrick D. Crispen teaches at the University of Alabama - Tuscaloosa
His Internet address is:
pcrispe1@ua1vm.ua.edu

The 68HC11 Microcontroller Family a Best Seller

Motorola Press Release

In July 1996 Motorola's 68HC11 8 bit microcontroller (MCU) family, one of the most widely used electronic devices in the world, will be twelve years old. First sampled in 1984, the 68HC11 was initially designed into a Chrysler engine control system, Canon EOS automatic 35mm camera, Delco Electronics Corp. engine control module, and Scientific-Atlanta satellite descrambler system. The architecture was flexible enough to meet the needs of many applications requiring 8-bit performance and was soon designed into a Conner Peripherals disk drive, Fisher Rosemount digital voltage meter, Ford instrument cluster and a variety of Motorola cellular telephone models.

The 68HC11 was the first architecture to offer on-chip Electrically Erasable Programmable Read Only Memory (EEPROM). Today, there are over 60 derivatives, with a variety of on-chip memory options, peripherals, speeds, voltages and packages. Motorola has shipped over 400 million 68HC11 MCUs to date (July 1995) and demand continues to grow.

"The 68HC11 microcontroller family is a powerful, robust architecture that offers high performance and integration to embedded system developers," said Greg White, 68HC11 operations manager for Motorola's Advanced Microcontroller Division. "Motorola's first 68HC11 customers continue to design with it, and we are continuing to develop new 68HC11 derivatives to serve our customers for many years to come."

68HC11 Versatility

The popular 68HC11 family powers a wide range of electronic products, including cellular telephones, robotics, computer peripherals like disk drives and modems, cameras and camcorders, cable set-top boxes, pagers, home security systems, automobile instrumentation, anti-lock brakes and airbags.

The 68HC11 architecture was originally designed in cooperation with Delco Electronics, a subsidiary of Hughes Electronics Corp. Delco Electronics first used the device in its model year 1987 engine control modules. Today, the 68HC11 is found in many Delco Electronics applications, including airbags, anti-lock brakes, vehicle electronics, instrumentation, and suspension, as well as engine and transmission control.

"In 1985, Canon developed its best-selling EOS SLR camera using Motorola's 68HC11 microcontroller because Motorola provided the design expertise, service, quality and superior engineering we required," said Shoji Ichikawa, Group Executive of the Photo Products Group of Canon. "We still use the 68HC11 at the heart of our newest EOS SLR cameras and are pleased to continue working with Motorola."

"Scientific-Atlanta has used Motorola 68HC11 microcontrollers in its products for more than 10 years. Scientific-Atlanta first designed the 68HC11 into its 1985 satellite descrambler system," said David Levitan, vice president and general manager of Scientific-Atlanta's Subscriber Systems. "Today, the 68HC11 acts as the nerve center for our flagship 8600x set-top terminal, offering all the performance and functionality the 8600x's advanced features require."

The Motorola Cellular Subscriber Group was also an early adopter of the 68HC11 architecture. Originally designed into their MiniTAC model of cellular telephones in 1986, today the 68HC11 is still used in many models, including the best-selling MicroTAC Elite personal cellular telephone — the world's lightest cellular phone at just 3.9 ounces.

68HC11 Innovation

Motorola, the leading supplier of microcontrollers with 19 percent total microcontroller market share and 30 percent 8-bit microcontroller market share (Dataquest: June, 1995), has forged the way for 8-bit microcontroller technology in many areas. At introduction, the 68HC11 pushed 8-bit

microcontroller performance into the 16-bit performance level with its 16-bit timer, accumulator and index registers. It was the first 8-bit microcontroller with on-chip EEPROM and the first to combine EPROM and EEPROM technologies on the same chip. In response to customer and market requirements, Motorola will continue to increase integration, performance, and versatility with features such as memory management, co-processors, LCD drivers, extended operating voltage ranges, low power operation, higher frequency operation, and advanced packaging.

CoCo OS-9 C Compiler "Problem" R. Gault, C. Simpson, C. Burke

CoCo programmers discover some idiosyncracies in Microware's K&R 6809 compiler

From Robert Gault:

I have just finished tracking down a very elusive bug in a 687 line C program. Just in case I can save you quite a bit of wasted time on your own programs, here is the problem I found.

I am sure that just about everyone has used the following code at some time: X=PEEK(memory)*256+PEEK(memory+1). It is the only reasonable way to get a 16bit number in memory into a variable. Most of my major programs are in assembly where it is even easier; just LDX memory. However, one of my projects required the C language to place a 16bit number from a character array into an integer variable.

Chet Simpson:

I find it interesting that any C programmer would use PEEK to access a memory location. Since this varies widely from machine to machine (i.e. -- on the IBM/PC you would use peek(segment, offset)) to access it in segmented memory it's never a good idea to use.

Robert Gault:

I thought, "well that's simple, I'll just use the routine from BASIC." So, I tried: Offset = buffer[0x1e]*256+buffer[0x1f]. This looks reasonable and it actually worked for some time, until the fatal day that I added an extra drive to my system. My program stopped working but with 687 lines of code, I did not give the above a second glance. Boy, isn't hindsight wonderful!

The above code gets PD.DTB, the drive table address, from a path descriptor. The added drive brought the value of buffer[0x1f] greater than $80. Disaster, and just because James McCosh liked SEX in his programming (for those who don't know, there is a 6809 m/l operator called SEX... editor).

The Microware 6809 C compiler sign extends all character values into integers before any math is done. This means that any value over $80 becomes $FF80+, ex. $C0 becomes $FFC0. So, lets say that the equation should be: offset= $52*256+$C0 or $52C0. The Microware compiler converts it to offset= $52*256+$FFC0 or $51C0. Even after I knew the equation was not getting the correct answer I did not see why for several hours!

Chet Simpson:

Since I'm not too familiar with the errors that lie within the Microware 6809 C compiler, it might be a good idea to change to a pointer reference instead of using peek (and if the compiler supports it, use typecasting to ensure that an unsigned integer value is used). For those wondering what I'm talking about, here is an example:

```
/* define variables */
unsigned char *buffer;
/* define pointer to buffer */
unsigned char byte;
unsigned int word;

buffer = 0x00;
/* set point to start of memory */
byte = buffer[memory];
/* get a byte from buffer memory */
word = ((buffer[memory]*256) +
(buffer[memory+1]));
/* get word from buffer memory */
```

Of course this does not always work on systems such as the PC which implement protected memory as memory location may be the start of the applications processing space.

(Note that Robert admitted to being primarily an assembly and BASIC programmer. This turns out to be important! He used a PEEK because that is what he is most familiar with, even though it is not common practice in C programming. As Chet points out, the PEEK solution would only work on a CoCo. Such machine limited commands aren't usually used in C because the code is written to be transportable to as many systems as possible with few changes to the source code -- editor)

Chris Burke adds:

I think the problem on the CoCo is that the type "unsigned char" is not supported. The CoCo uses a pre-ANSI K&R subset compiler. Since char is aways signed on the CoCo, the compiler is correct to sign extend the value; a C compiler is *required* to extend char to int during intermediate calculations. One correct coding on the CoCo would be:

```
#1: Offset = buffer[0x1e]*256 +
(buffer[0x1f]&0x00ff)
```

I prefer (on the CoCo):

```
#2: Offset = *(int*)(buffer+0x1e)
```

BUT this assumes that sizeof(int) is 2 - on many UNIX systems, sizeof(int) is 4. Also, #1 assumes Motorola-style byte ordering; #2 assumes that the data being read was originally written usir the same byte ordering as the read. #2 assumes that there's no catastrophic result to reading non-byte data from an odd address (i.e. don't try this on a straight 68000, 68306, or 68070 unless you're sure the data is properly aligned).

Another useful formulation, which avoids the odd address problems of #2 and is on some machines faster than #1 is:

```
#3: Offset = (buffer[0x1e] << 8) |
(buffer[0x1f] & 0x00ff)
```

....or the pointer-based equivalent. If you have unsigned char, you can eliminate that pesky & 0x00ff step.

Basic09 In Easy Steps

Chris Dekker

RunB operators, strings, and control structures

In this article I will finish the discussion of RunB. The topics: operators, strings and control structures.

Operators like =, < and > are supported by different types of subroutines inside RunB. These routines deal with byte/integer, real number or string type variables. The subroutine for bytes/integers looks like this:

```
LDD 7,Y
SUBD 1,Y
BLT true
BRA false
```

As you can see, there isn't much to it. You substract the two variables and compare the result to zero. This code is executed when your program reads like this: IF x<y THEN ... "true" and "false" are labels. They point to other subroutines that set the value of a boolean variable to true or false. This variable is returned to RunB, which then continues program execution based on the variable's value.

BLT is a 6809 mnemonic that means "branch if lower than". To see if it should branch to "true" here, the 6809 checks two flags in the condition code register. (Yes, the same regs.cc we check for errors after executing a system call.) If the two flags differ, the processor starts executing "true". If they are the same (both set or cleared), it executes the "BRA false" instruction.

RunB contains an entire battery of such routines: one for each operator or combination of them (e.g. <> or <=). The only difference between these subroutines is that the "BLT" instruction is replaced by another instruction. This causes the microprocessor to check other (sets of) flags in the condition code register. [If you're not familiar with ML programming: the SUBD instruction automatically sets/clears these flags depending on the result of the substraction.]

There are similar sets of subroutines for strings and real numbers. The difference being that you can not substract strings or real numbers in the same way as integers: the routine for real numbers compares two values (which is quicker then substraction), while strings are compared on a character for character basis.

Strings are actually compared by comparing the ASCII values of the individual characters. If they are the same, the next set of characters will be compared. If not the condition code flags are set according to which of the two characters is the smallest and processed in the same way as decribed above.

This works great except that it makes the comparison case sensitive. If you want to compare strings without distinguishing between upper and lower case, you're out of luck. You will have to do a conversion in Basic (which is slow), do the comparison in Basic (even worse), use the Compare system call (takes some setting up to stay out of trouble) or write your own ML subroutine.

Now that we are on the subject of strings: What's faster: LEFT$ or RIGHT$? Or is there no difference? Actually there is and LEFT$ is faster, although you will need a very long string to actually notice the difference.

The reason is quite simple: under Basic09 all strings start in the leftmost position of their alloted space. Although this seems logical enough to us, it means that RIGHT$ (and MID$) have to do an extra copying operation to shift the portion of the string they have to return to the left. By the way, MID$ will jump to LEFT$ if you specify 0 as the starting column.

LEFT$ works as follows: it calculates the new end position of the string. If this is beyond the current end, it leaves the string alone and exits. Otherwise it will copy $FF (Basic09's string delimiter) to the position right after the new string end. It also updates some pointers. All of this takes anywhere from 15 to 40 MPU cycles. Of course you have to add to that the overhead of copying strings to and from the expression stack. This adds at least a few hundred cycles.

RIGHT$ and MID$ have (besides abovementioned copying) some extra checks to make. RIGHT$ should generally take 50 to 100 MPU cycles and MID$ 100 to 150 cycles. TRIM$ is also a very short routine that can be compared with LEFT$ in speed (unless you have lots and lots of spaces to trim).

SUBSTR can run in a 100 or so cycles, but may also stretch into the thousands. This entirely depends on the strings you pass to it. That leaves us with VAL and STR$. I won't even try to guess how long it takes for these functions to execute. This is due to the variety of variables these routines have to deal with.

VAL and STR$ both enter large subroutine packages (approx. 200/160 lines respectively). Both also call other subroutines inside loop structures. So, if you want to write a speedy program, don't use them more often then absolutely neccessary.

The same goes for the, innocent looking, print USING statement. Believe it or not but the subroutines associated with this command make up close to 10% of the RunB module. Now, you won't need all that code to print a single space; but the overhead associated with PRINT USING compared to a simple PRINT is substantial.

Personally, I use PRINT USING only to create decent looking tables. For that application you need complete control over positioning or the results may not be something to brag about. Other than that:

```
PRINT " ";var   and
PRINT USING "x1,i2";var
```

have the same result but the latter is quite a bit slower. Now, last but not least: control structures. These includes statements like GOTO, GOSUB, IF/THEN, LOOP/ENDLOOP, etc. To start with the easiest of all: GOTO.

RunB takes this quite literal. The code implementing GOTO looks like this:

```
LDD ,X
ADDD modstart
TFR D,X
RTS
```

A little explanation might be in order. To make this work Basic09's assembler converts the line number in a GOTO statement into an offset relative to the starting address of the module. This address is loaded into the variable "modstart" when RunB starts executing a module.

All RunB has to do when it encounters a GOTO statement is: add the offset to the starting address and transfer the result to the X register. The X register is used as pointer. It always points into a packed Basic09 module, so RunB knows where the next instruction or variable can be found.

Why all this fuss about GOTO? Well, first of all it shows how fast these jumps get executed. The above code takes 23 MPU cycles (17 cycles for a 6309) and has very little overhead: just looking up the postion of the GOTO subroutine in a jumptable and entering it which takes 25-30 cycles.

The second reason is that RunB uses this

code a lot. This must be a hair raising experience for people who insist that any program with a GOTO statement is inferior, but it's true. For instance ENDLOOP is literally translated into: GOTO LOOP. ENDWHILE, ENDEXIT and ELSE also jump to this subroutine. Control statements like EXITIF, UNTIL and NEXT branch to this routine if a certain condition is not met.

GOSUB (and RETURN which almost executes GOSUB backwards) are a little more complicated then GOTO because they have to preserve the contents of the X register (so your program knows where to return back to) and check the stack for over/under flow errors. Nevertheless the job gets done in about 50 MPU cycles. This beats starting a new module by a mile (or two). So don't get in the habit of converting each 5 line subroutine into a separate module.

Creating your own jumptable with ON x GOTO/GOSUB adds at least a few hundred MPU cycles, but is still a lot faster then going through an entire series of IF/THEN statements. This is due to the fact that every IF is followed by a jump to the routines I described at the start of this article. The THEN (and ELSE) statements have been replaced by offsets that are processed by the GOTO routine.

There is one set of program flow controls whose code looks entirely different than that of the others. I am talking about FOR/NEXT/STEP. Although easy for us: what could be simpler then FOR i=1 TO 10 ... NEXT i?; it is by far the most complicated control structure for Basic09.

The code even has it's own jumptables because it comes in 4 variations: integer (step 1 & step x) and real (step 1 & step x). Having said that, Basic09's assembler does an excellent job setting things up so there is probably little difference in execution speed with other loop structures. My best guess is that WHILE/ENDWHILE is somewhat faster if you can use an implicit counter (one that is already used for another purpose). However if you have to define and increment a separate counter, FOR/NEXT seems somewhat faster. Anyway the difference is too small to make much difference for your program's performance.

So,, there you have it!! Now that we all have some idea why certain programs zip along while others crawl; you, too, can be on your way to writing that impressive program you always dreamed about.

1996 Chicago CoCoFest Vendors

Alan Dages
Used CoCo items
404-469-5111

Adventure Survivors
Adventure game newsletter and programs
24 Perthshire Drive
Peachtree City, GA 30269

Chicago Area OS-9 Users Group
10 McCarthy
Forrest Park, IL 60466

ChiCoCo BBS
BBS number 312-735-3355

Digigrade Productions
DAS MM/1 CD adventure game
1325 West Lincoln Hwy, Apt 412A
DeKalb, IL 60115

FARNA Systems
CoCo & OS-9 magazine, books, some software
Box 321
Warner Robins, GA 31099

The OS-9 Users Group, Inc.
6158 W 63rd Street, Suite 109
Chicago, IL 60638

Wittman Computer Products
AT306 and MM/1 computers, hardware, and software
39 South Lake
Bergen, NY 14416

Northern Exposure
NitrOS9 CoCo OS-9 speed-up package, other CoCo, OS-9, OSK software
7 Greensboro Crest
Ottawa, K1T 1W6 (Canada)

R.C. Smith
Used CoCo items
Phone 404-469-6601

Ron Bull
Used CoCo items
115 Ann Street
Duncannon, PA 17020-1204

SBUG
New and used CoCo items, supplies
1251 West Sepulveda Blvd, Suite 400
Torrence, CA 90502-2677

Strongware
New MM/1 and CoCo software
Box 361
Matthews, IN 46957

The CoCo Cellar
Used CoCo items
735 Praireview Lane #3B
Woodstock, IL 60098

Glenside CoCo Club
CoCo support newsletter, club, fest sponsor, some used items
1209 Winslowe Drive, Apt 103
Palantine, IL 60067

Hawksoft
CoCo and MM/1 software, CoCo hardware
28456 S.R. 2
New Carlisle, IN 46552

John Brown
Hazelwood OS-9/68K, EURO-K, and TC-70 computers and software
7887 Longview Drive
Bryan TX 77808-8762

Justin Time
Used and new CoCo soft/hard ware
Box 82
Glen Ellyn, IL 60138

Monk-O-Ware
Used items, some new hardware and software (2MB expansions for CoCo3)
Box 1903
Racine, WI 53401

Motorola Licenses Macintosh Operating System

The Motorola Computer Group (MCG) announced in February 1996 that it had reached an agreement with Apple Computer, Inc. (Cupertino, CA) to license the Mac(tm) OS operating system. This licensing agreement applies to version 7.5.x and includes access to the next major release of the operating system. With this the Motorola Computer Group also becomes the first licensee to be able to sub-license the Mac OS with its motherboards and private-label systems to the OEM marketplace. MCG will use this license to increase Mac OS presence in the markets it serves which will offer users more supplier choices and also offer software developers with incremental market opportunities. Motorola plans to exploit the PowerPC Platform, (also known as the Common Hardware Reference Platform (CHRP)) announced by the AIM (Apple, IBM, Motorola) alliance at Comdex last November, to bring its offerings to the marketplace.

Under the agreement announced today, Motorola will sell, support and service Mac OS-based PowerPC (tm) Platform desktop systems, positioning MCG as a premier supplier of PowerPC platforms. "This licensing agreement is strong evidence that Motorola is committed to PowerPC and the Mac OS market. It complements our plan to support PowerPC-based multiple operating solutions to the market," said Joe Guglielmi, corporate vice president and general manager of the Motorola Computer Group. "We applaud Apple for its open vision. This agreement greatly expands the choice of Mac OS-based system suppliers worldwide, which will allow more individuals access to this significant platform. We are also excited about the opportunity to sub-license this technology to OEMs that want to provide Mac OS applications which expand their markets." "We are extremely pleased with Motorola's commitment to the Mac OS," said Dr. Gil Amelio, chairman and CEO, Apple Computer, Inc. "Motorola's reputation for manufacturing high-quality products, its modern manufacturing facilities, and the fact that it will be able to sub-license the Mac OS, opens a new range of opportunities to computer manufacturers purchasing motherboards and systems from Motorola. This should result in a significant growth of the Mac OS platform."

In support of this agreement, Motorola intends to leverage its Six Sigma manufacturing capability to provide a range of low-cost, multiple operating system solutions based on the PowerPC Platform to support demand for the Mac OS. By combining the Mac OS-based desktop system with its Windows NT(tm)-based server product line, MCG can provide world class client server solutions. Motorola plans to distribute its Mac OS-based products to the enterprise marketplace worldwide by leveraging its existing corporate reseller, aggregators, and system integrator distribution channels. MCG is also planning to add new corporate resellers that focus on solutions for the enterprise marketplace.

MCG will use its recently-announced joint venture with Panda Electronics Group in China to distribute Mac OS-compatible systems to that rapidly expanding marketplace. Nanjing Power Computing LTD will use Panda Electronics distribution channels to distribute Mac OS-based desktop systems to consumer and education markets in China. In addition to this joint venture between MCG and Panda Electronics, Motorola has established partnerships with several organizations in the People's Republic of China to provide software development and localization support.

Motorola's Semiconductor Products Sector (SPS) fully supports a broad-based family of scalable, high-performance PowerPC microprocessors for the Mac OS, Windows NT, AIX (an IBM version of UNIX), and Solaris operating environments. In addition to the processor family, SPS provides comprehensive support for building computer solutions based on the PowerPC Platform including system and software development tools, compatible chip sets, reference design systems and open firmware. Motorola SPS will expand on these offerings to include support for the Mac-compatible market.

http://www.mot.com/computer

Editors Comment:

What this really means is that Motorola hopes to do as Intel does... make motherboards. These will be made available to any computer manufacturer. So it may be that PowerMac clones start coming around soon. But don't expect the price to be much lower than Apple's for a while. Unless someone starts pumping them out wholesale, they won't make much impact on a shrinking market.

Face it, for better or worse (I think worse!), the Intel based systems are slowly killing everything else off! The only thing that keeps the government off Intel's back (for a monopoly) is the fact that AMD and Cyrix make clones of the major Intel chips. If Intel is smart, they won't interfere with those guys to much either... keeps Uncle Sam off, you know!

I viewed the upgrade to PageMaker recently (Adobe PM 6.0). The hardware requirements are atrocious! It will run on a 486... with 10MB of free space (meaning a minimum of 12MB if you run nothing but PM6 and Windows) and 90MB of hard drive space for a full installation... 50MB REQUIRED for VIRTUAL MEMORY! As if 10MB free weren't enough! And a Pentium is recommended! This is unreal! There aren't that many obvious extra features in PM6 (as compared to PM5), and PM5 runs great on my 486 with only 8MB or RAM, and then I even have several programs loaded in memory! And to top that off, PM4 will even run adequately on a 4MB 286/20MHz machine (a[illegible] you're just a tad patient)! What happened? They had to make [illegible] patible with Win95.

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

HRSDOS - $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 - $20.00
Access your SmartWatch from DECB! Adds function to BASIC (DATE$) for accessing date and time. *Only $15.00 with any other purchase!*

RGBOOST - $15.00
Make the most of your Hitachi 6309 by using it with DECB! Uses 6309 functions for up to 15% gain in speed. Compatible with all programs tested to date! *Only $10 with any other purchase!*

Robert Gault
832 N. Renaud
Grosse Pointe Woods, MI 48236
313-881-0335
Please add $4 S&H per order

Northern Xp[illegible]r

CoCo OS-9:
NitrOS9 - Upgrade OS-9 Level II for the CoCo to a new level of performance! Requires Hitachi 6309. Call or write for upgrade info. $29.95
Shanghai:OS-9 or **Thexder:OS-9**
Send manual or ROM Pak to prove ownership. Transfers and modifies code to run under OS-9! $29.95
Rusty - Launch DECB programs from OS-9! $20.00
SCSISYS 2.2 - SCSI hard drive drivers... fast! Supports 256 or 512 byte sector drives! $25.00
Hitachi 6309 CPU $15.00
SIMM 512K Memory Upgrade - runs cooler!
W/512K - $44.95 w/o memory - $39.95

OS-9/68000:
OSTerm 68K - v2.2terminal program. TTY/ANSI/VT100/KWindows support $50.00

7 Greensboro Cres
Ottawa, ON K1T 1W6
CANADA

All prices in US funds. Check or Money Order only.
Prices include S&H.

for all your CoCo hardware needs, connect with

CoNect
449 South 90th Street
Milwaukee, WI 53214
(pulland@omnifest.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 at 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!

ADVERTISER'S INDEX

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!

Wittman Computer Products

HARDWARE * SOFTWARE * CONSULTING

We Offer Gift Certificates

Software

Game Pack - Sea Battle, Minefield, KnightsBridge, Othello, & Yahtzee	$30 MM/1 $25 COCO
Variations of Solitaire - Pyramid, Klondike, Spider, Poker, & Canfield	$30 MM/1 $25 COCO
Gold Runner 2000 - KWindows game	$35 MM/1
KChess - GNU Chess with KWindows GUI	$25 MM/1
GNU Chess Source Code Disk	$5 OSK
CirCad - Circuitry CAD program with PostScript output	$80 MM/1
DeskTamer v2.0 - Personal Information Manager	$65 MM/1
LaTerm / LaDial - GUI terminal package with script auto-dialing	$65 MM/1
LaPhone / LaFax - GUI voice, fax & address database manager	$65 MM/1
InfoXPress - auto-dial and download messages	$75 OSK $55COCO/OS9
X-10 Home Controller - Program for monitoring your X-10 system	$40 MM/1
M6809 - CoCo OS-9 emulator	$65 OSK

TVP Point of Sale - Fully integrated multi-user accounting, inventory, ordering, and cash drawer. $CALL

Hardware

WCP306 single board computer with 16 bit PC/AT I/O bus, MC68306 CPU running at 16.67MHz (code compatible with 68000), OS-9 v3.0, MGR GUI, many utilities.
3 Slot board - **$400** 5 Slot board - **$425**

We also carry Point-of-Sale equipment, hard and floppy drives, cases and power supplies, data/fax modems, keyboards, memory, mice, and more!

For more details, write or call for a catalog.

Wittman Computer Products
39 South Lake Avenue
Bergen, NY 14416
716-494-1506

e-mail: ww2150@acspr1.acs.brockport.edu
(acspr "number 1" - lowercase l won't go through!)

For superior OS-9 performance, the

SYSTEM V

Provides a 68020 running at 25 MHz, up to 128 MBytes of 0 wait-state memory, SCSI and IDE interfaces, 4 serial and 2 parallel ports, 5 16-bit and 2 8-bit ISA slots and much more. The **SYSTEM V** builds on the design concepts proven in the **SYSTEM IV** providing maximum flexibility and inexpensive expandability. Optionally available at 33 MHz.

AN OS-9 FIRST - the MICROPROCESSOR is mounted on a daughter board which plugs onto the motherboard. This will permit inexpensive upgrades in the future when even greater performance is required.

G-WINDOWS benchmark performance index of a **SYSTEM V** running at 25 MHZ with a standard VGA board is 0.15 seconds faster than a 68030 running at 30 MHz with an ACRTC video board (85.90 seconds vs 86.05 seconds).

For less demanding requirements, the

SYSTEM IV

The perfect, low cost, high-quality and high-performance OS-9 computer serving customers world-wide. Designed for and accepted by industry. Ideal low-cost work-station, development platform or just plain fun machine. Powerful, flexible and expandable inexpensively. Uses a 68000 microprocessor running at 16 MHz.

G-WINDOWS a PROVEN WINNER

FOR OS-9

Available for the **SYSTEM IV** and **SYSTEM V** computers, the PT68K4 board from Peripheral Technology and the CD68X20 board from Computer Design Services. Resolutions from 640 x 480 x 256 to 1024 x 768 x 256. Support for multiple VGA cards running different processes, different portions of the same process or both.

G-WINDOWS is now AVAILABLE for OS-9000 from *delmar co*

Support for generic VGA boards and SUPER VGA boards including ET4000 (Tseng Labs), OTI067 (Oak), CT452 and CT453 (Chips and Technology), GENOA, WD90C11 (Paradise) and S3 (S3 Inc.) for modes from 640 x 480 to 1280 x 1024 depending on board.

Distributor of MICROWARE SYSTEMS CORPORATION Software

This ad was prepared and printed using QuickEd under OS-9.

delmar co

PO Box 78 - 727 North Broad Street - Middletown, DE 19709

302-378-2555 FAX 302-378-2556