

# OS-9 Newsletter

Volume IV Issue 10

Bellingham OS-9 Users Forum

October 31, 1993



The "Hot Rod" will utilize 2 HC63C09E chips. The Slave CPU will share some memory in the master CPU's area for information transfer. It will have access to a maximum of 10 ISA slots which the first slot will be dedicated to a VGA/SVGA video control card. The other 9 slots can house whatever IBM hardware cards you wish to add. Remember that OS9 will need device drivers and descriptors for each item you plug into this. We're doing a 3 fold improvement to the CoCo with the card. 1.) Access to higher graphic resolutions through the first ISA slot. 2.) Higher bus speed by removing the GIME entirely and replacing it with a daughter board which performs the exact same functions and provides the interface to the slave CPU. 3.) Faster processing speeds by utilizing 2 CPU's to manipulate the data. Throughput can theoretically be up to 10 MHz.

Things are moving at a rather fast pace and it looks like we'll have the first prototype done by December. After initial test run 3 more will be produced for Beta and Development. Wes Gale will be the recipient of one for some rather obvious reasons.

The new issue of "Computer Heaven" has just been completed. This issue contains the preliminary data for the "Hot Rod" card. It includes the preliminary block diagram and proposed memory map that may be used.

This is advanced information on the product and is subject to change as development continues. However we do not anticipate much severity in any future changes and expect most information listed to remain as stated in the diagrams and notes. I highly encourage programmers and hardware hackers to send for this issue. It is free information and all that is required to receive it is a SASE sent to the following address: Eight Bit Heaven; 1108 E. Lexington # C; El Cajon, CA. 92019. For those who reside in Canada send \$1 US so that I can purchase stamps and envelopes for mailing.

==Shaun Marolf==

FidoNET:OS-9 Echo

## The Rocket

— more details —

Microware's new pricing made it significantly less attractive for Burke & Burke to enter the OSK hardware market with a product which would ship only 100 units. There are some real advantages to the new pricing, but in our case it essentially doubled our production costs.

Depending on our ship date, we might have been able to get the older Microware pricing and product. Rocket orders never reached the 100 unit mark, so Burke & Burke couldn't commit to an order. Microware was both flexible and generous, but the deal just didn't work out.

In the meantime, my primary employer stepped up my travel schedule and duties to a level that severely limits the amount of time I have to develop software at Burke & Burke. The most viable Rocket compromise was one in which we purchased less Microware software, and built more "work-alikes" (e.g. our own disk utility set DIR, DEL, etc.). This, too, didn't work out because I had no time to develop the software.

Several people on the CoCo List have volunteered to work on software for The Rocket. If an established hardware vendor were to build the circuit board and ship it with the OS9 kernel, perhaps the community could piece together the rest of the tools and O/S as a User Group project. If any vendor or a user group is interested in this, please contact me and we can discuss transferring the circuit design for *The Rocket*.

==Chris Burke==

Burke & Burke

## \*\*\* IN THIS ISSUE \*\*\*

The "HOT ROD"	Pg. 1
Double CPU daughter board for 10MHz, SVGA CoCo	
<i>The Rocket</i> by Chris Burke	Pg. 1
More information on it's demise, but possible revision.	
OS9 Community Network BBS Node List	Pg. 2
FidoNET Listing of official OCN Bulletin Boards	
The Development Process	Pg. 3
A dialogue between Boisy Pitre, Jesse Oberreuter, Chris Burke	
Correction The "Correct All" Fix for the CoCo 3	Pg. 4
MM/1 Upgrade by Zack Session	Pg. 5
Report on new 68070 Upgrade	
MGA on KiX by Frank Hogg	Pg. 5
Report on MGA graphics for both Kix 20-30 machines	
Development of new extensions to Level II	Pg. 6
by Alan DeKok	
Questions and Answers	Pg. 7
What is CoCo List? / Level II vs. OSK / Using BINEX	
Telecommunications on a CoCo using SuperComm	Pg. 8
by Rick Uland	

# OS9 Community Net Work

## BBS NODES

The following is a list of FidoNET Bulletin Boards that are "official" OCN Nodes. The SysOps (System Operators) are OCN Librarians and/or Regional Officers of the OS9 Community Network. These Boards maintain a complete library of Public Domain Software submitted to the OCN. Files available also include OCN Reports and Minutes. If you have any questions or wish to join the "OCN", contact the SysOps at on the of FidoNET nodes listed below. There is no membership fee.

Node address: 1:18/75 SysOp name: Dave Spicer  
BBS Name: Access CoCo of L.A. Phone Number: (205) 598-2100

Node address: 1:106/941 SysOp name: Terry Goode  
BBS Name: The Golden CoCo Phone Number: (713) 941-1542

Node address: 1:133/510 SysOp name: Newton White  
BBS Name: ACS BBS Phone Number: (404) 636-2991

Node address: 1:154/888 SysOp name: Kerry Kowalski  
BBS Name: Data Stash Phone Number: (414) 684-4115

Node address: 1:163/519 SysOp name: Yves Souliere  
BBS Name: Discus Phone Number: (819) 771-3792

Node address: 1:202/343 SysOp name: Warren Brach  
BBS Name: Ocean Beach Phone Number: (619) 224-4878

Node address: 1:202/617 SysOp name: John Reece  
BBS Name: CoCo Exchange Phone number: (619) 272-3643

Node address: 1:202/624 SysOp name: Jim Harrison  
BBS Name: The Byte Box Phone Number: (619) 277-4618

Node address: 1:250/610 SysOp name: Ken Patience  
BBS Name: Whitelightning Phone Number: (416) 469-2681

Node address: 1:264/211 SysOp name: Doug James  
BBS Name: No Baudy Home! Phone Number: (804) 744-9260

Node address: 1:282/102 SysOp name: Jim Sartian  
BBS Name: BB's Place Phone Number: (612) 869-7752

Node address: 1:321/312 SysOp name: Brian Stewart  
BBS Name: The CoCo Workshop Phone Number: (413) 593-3944

Node address: 1:345/200 SysOp name: John Wight  
BBS Name: The CoCo Library Phone Number: (808) 735-3776

Node address: 1:346/9 SysOp name: Dennis Mott  
BBS Name: Data Warehouse Phone Number: (509) 325-6787

Node address: 1:359/251 SysOp name: Ken Flanagan  
BBS Name: Pot O' Gold Phone Number: (604) 564-8869

Node address: 1:382/107 SysOp name: Tim Jones  
BBS Name: Trial Run Phone Number: (512) 280-6578

Node address: 1:396/27 SysOp name: Gene Clifton  
BBS Name: The Node III Phone Number: (504) 347-4320

Node address: 1:2613/331 SysOp name: Erik Seielstad  
BBS Name: MultiMediaCircus Phone Number: (716) 637-2361

Node address: 1:3403/3 SysOp name: Mark Johnson  
BBS Name: Columbia Heights Phone Number: (206) 425-5804

— MAROLD KISTNER —  
FidoNET:OS-9 Echo

## OS-9 Newsletter

Editor: Rodger Alexander

*OS-9 Newsletter* is published monthly by the **Bellingham OS-9 Users Forum** and is protected under United States Copyright Laws. No material may be reproduced or copied in whole or in part without the expressed written permission of the Bellingham OS-9 Users Forum, 3404 Illinois Lane, Bellingham WA 98226

*Submissions* are welcomed in any format and can be mailed to the above address or sent via electronic mail to the editor: Rodger Alexander, on Delphi (UserID: SALZARD) or FidoNET (1:301/3401@fidonet.org) or Internet (ralexander@bones.nwrdc.wednet.edu). Unfortunately, we do not have funds to reimburse authors of selected articles. However, a complimentary copy of the *OS-9 Newsletter* containing your article will be mailed to you, PLUS the satisfaction that you will have the admiration and appreciation of all of our readers.

The *Bellingham OS-9 Users Forum* is a hobbyist club, organized for the purpose of providing information, services, products and events that support the OS-9 operating system for 6809/68xxx based computers. Our efforts are not intended to earn or generate any profit for the club or any of it's members.

### TO SUBSCRIBE

For 12 monthly issues of the *OS-9 Newsletter*, please send a US check or money order for \$12 or \$7 for a 6 month subscription. Mail your subscription order to:  
OS-9 Newsletter  
3404 Illinois Lane  
Bellingham, WA 98226

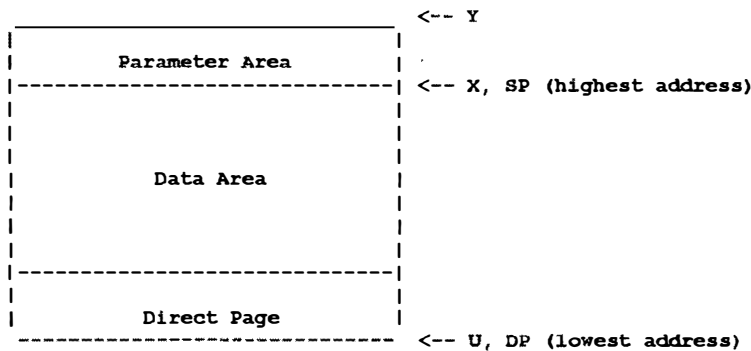
Include your name, address and telephone number. You will receive your *OS-9 Newsletter* no later than the 10th of each month. Canadian orders, \$13.50 for 1 yr. or \$7.60 for 6 mo. Foreign orders \$18 for 1 yr. or \$10 for 6 mo.

# The Development Process . . . . .

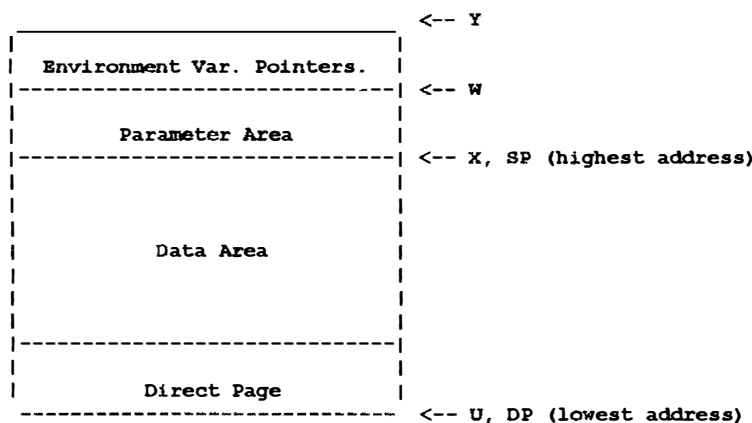
A Dialogue between Level II Developers: Boisy Pitre, Jesse Oberrueter and Chris Burke

After doing some research on the feasibility of implementing environment variables under OS-9 Level II, I'm close to proposing a strategy. It would require patching F\$Fork, and adding additional system calls (F\$GetEnv, F\$SetEnv). It would also only work on a 6309 OS-9 system.

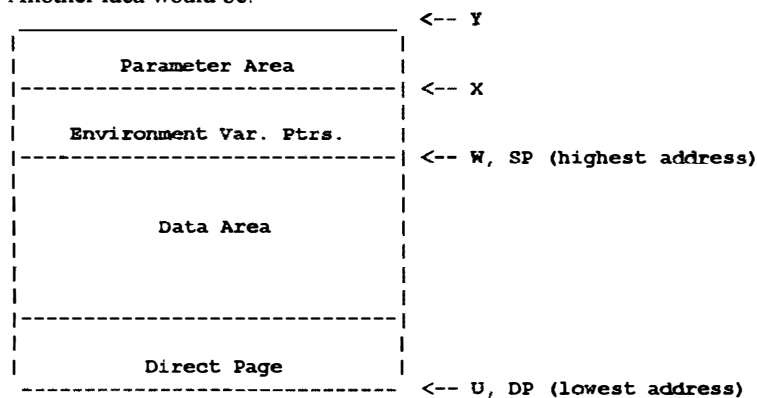
Here's the register convention upon entry of a process:



Here's the proposed register convention with environment variables:



Another idea would be:



Both methods have the potential for break existing programs, but I really can't see a cleaner way around it.

W would point to a set of pointers to environment variable strings.

F\$Fork would need to be patched to do the following:

1. Allocate memory for pointers and strings using the parent's environment size.
2. Copy the strings from the parent's data area to the child's, and reset the child's environment pointers. It would also be necessary to store the pointer to the head of the pointers list inside the process descriptor.

F\$GetEnv and F\$SetEnv would be provided as a kernel extension (OS9P3). Both would manipulate the environment pointer list as well as environment strings.

The underlying question in all of this is "How does OS-9 allocate memory in the process' data area and how would it interfere with the expansion of environment variables for each process?"

For instance: When the F\$Mem system call is invoked, does it move the parameter area up or down as needed to expand/contract the data area? I don't readily know the answer to this question, therefore I cannot draw a conclusion as to how the register conventions after a fork should look like. Any suggestions would be helpful as I continue to hammer out the details of this implementation.

Boisy G. Pitre

Microware Systems Corporation, Des Moines, Iowa

Internet: [boisy@microware.com](mailto:boisy@microware.com)

**Hmmm, interesting proposal.**

You are effectively splitting up the system to allow more system RAM per device. I take it this is a precursor to Virtual Memory? In theory, I don't see why you couldn't do this. In fact, you could probably set things up so that each device could potentially have its own map. Then you certainly wouldn't run out! Of course the 64k limit is always going to be w/you.

It is still possible to do Virtual Memory on the CoCo. Effectively, when a task switch occurs, the MMU gets remapped to place whatever processes you're switching to (and anything else it is linked to) into memory. Since this memory is normally floating outside of the 64k map, there is no reason why it couldn't be on disk. Doing this sort of thing would actually be

casier under Level II than OSK because processes don't expect to see a flat map; they and their linked friends are the only ones in visible memory. This really wouldn't be too difficult -- I think the worst part would be using the Hard Drive. This would be really painful in system state...

Hey! You could set up a dameon that would be responsible for cycling the processes between disks. During low user interaction periods, it could just cycle through active processes based on their priority. When something actually needed preempting (or waking up because you wanted to use it), the task switched could pass the request on to the dameon, switch to the dameon, and go back to what it was doing. When the dameon completed, it could toss that process back on the active list. The Next process call would need to be fixed to check a list of processes in RAM, and only activate those that were there or needed to be there. Since the Dameon would handle all swapping, it could check a bit in a module header to determine whether the process could be swapped out or not, so a terminal program (or itself) couldn't be dropped! (Wow -- cool idea Jess; that would save a LOT of kernel rewriting..) With the addition of a system call to tell the kernel that the dameon process was running, you would boot up normally, and somewhere in the startup script (or CC3Go if you wanted to get fancy), you would do something like VM /dd -b=1024k -t=10 &

Jesse Oberrueter (KB7PSG)  
joberreu@seattleu.edu

#### How about a RAM Disk?

Might be better to allocate environment variables using 8K blocks which ARE NOT part of the system or process map, as in a RAM Disk. Then store a 24 bit pointer to a "directory" of environment variables in the process descriptor. Then to read environment variables, you need:

```
F$GetAllEnvSize
pointer=malloc(size_of_env)
F$GetAllEnv(pointer) (returns a data structure similar to
your proposal)
```

This would work on any processor type.

To change an environment variable, use:

```
F$SetEnv(name_ptr,value_ptr) (assuming null-terminated
variable value and name...and automatically defines
undefined vars.)
```

To read an individual environment variable, use:

```
F$GetEnvSize(name_ptr)
pointer=malloc(size_of_var)
F$GetEnv(pointer)
```

The brazen could skip the 1st 2 steps and allocate a fixed size buffer I suppose. Or the interface could be changed to accept a maximum length, and return the actual length. In any case, you're not confined to the 6309, and you have a large space available for environmental variables. F\$Fork copies the calling process' directory and values for the child; F\$Exit

dallocates. You could also take a cue from the Mac, and use handles (\*\*, not \*) in the data structures. This would allow you to do garbage collection as the environment space becomes fragmented.

Here are a couple of more ideas:

If the TOTAL environmental space for all environmental variables in all processes is limited to 64K, the env. var. pointer would only be 16 bits and would be an offset into a "DAT image" for env. vars.

Another option is to use a true RAM disk device called /env. Sub directories for each process called /envxxx, where xxx is the process ID. Files in each sub directory called <name>, containing <value>. Files are read-write by owner. F\$Fork creates /env/envxxx, copies all files from /env/envyyy (the parent). F\$Exit blows away /env/envxxx. Program uses standard file I/O calls to read, create, and modify env. vars. Less efficient, but pretty simple - and in theory, the name of the path for "/env" could be in the INIT module, allowing any existing device to be used for env. var storage (e.g. /d0/sys/env could be used, if desired, but this might slow down FORK a lot!).

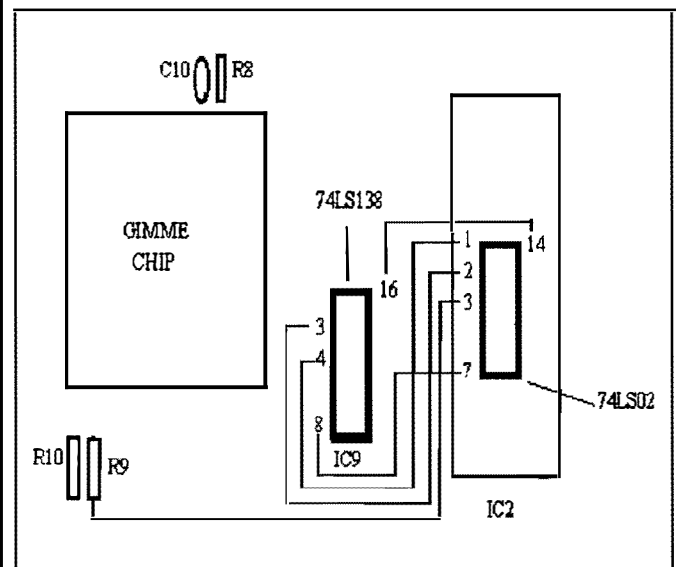
Chris Burke

"Chris Burke <burke@MDD.COMM.MOT.COM>"

## Corrections

In the "Correct All" Fix for the CoCo-3 article (July '93), figure 3 on page two shows a connection between pin 3 of the 74LS02 and the "top" end of R9. I wired my CoCo's as shown in this digram and the circuit appeared to work fine on my machines. However, **Buzz Jones** pointed out that the junction between C10 and R9 is actually located at the "bottom" of R9.

### Corrected diagram



## 68340 Upgrade to MM/1

## MGA for KiX

I am the proud owner of an MM/1A, that is an MM/1 which has been upgraded with the 68340 Accelerator Upgrade. **What it is:** A "daughter board" which has the 68340 and a few support chips on it. Two replacement ROMs. One replacement PAL. Floppy disk with new kernel and drivers.

Cost: \$325

### What is the 340 upgrade suppose to give you?

**More Power.** The 68070 was a backwards engineered 68000 and therefore had numerous situations where the micro code was less efficient than an original 68000. The 68340 is a genuine Motorola product and is an improvement over the 68000 in micro code efficiency. Most instructions take fewer (usually half as many) clock cycles to complete plus the 68340 has some of the 68020 instructions. Even though the "normal" clock speed of the 68340 will be 16.59 MHz for most MM/1 users, it still rates about twice as fast as a 15MHz 68070. According to the upgrade docs, the 68340's clock is adjustable from 11.98 MHz to 25.80 MHz (although one user has reported he was able to set his down to 8 MHz with a special system state program.) If you have slow DRAMS (i.e., 100 nanoseconds) you will only be able to reliably run at about 12 MHz. 80 nanosecond DRAMS will allow a clock speed of about 16 MHz. The docs say that 60ns DRAMS should allow a clock speed of 20 MHz, but Kevin Pease told me that all things considered (other chips on the boards, etc.) that one really wouldn't be able to get reliable operation above the 16.59 MHz value.

**Better system IO** since the DMA transfers are not limited to a size of 64K. The new SCSI driver which comes with the upgrade apparently takes this new feature into consideration.

**More and better serial ports.** The 68340 has three serial ports, being used as /t0, /t1 and /t5. These are improvement over the '070s two serial ports in that they look more like 68681 ports, they are full ports with hardware handshake, but CD is not currently implemented. A standard /t3-/t4 paddle is required to use /t5. (The header for it is on the

### Conclusion:

All in all, these advantages add up to a MM/1 which is roughly twice as fast as it was before the upgrade. All of the "benchmarks" I tested held true to this. I feel that the investment was well spent. I am very pleased with the speed increase in my MM/1 and can live with a few minor bugs which have not been solved yet. I would have really liked it if there had been SOME documentation

==Zack C Sessions==

CoCo - Tandy Color Computer List  
<COCO@pucc.Princeton.EDU>

All KiX computer systems have a 32 bit expansion bus that runs at full CPU speed. This is noteworthy when you consider that the common IBM PC type computer's AT bus is only 16 bits and runs at only 8 Mhz. The bus in a 33Mhz KiX30 is 8 times faster than a similar system using an AT bus. To overcome the limitation of the slow AT bus the PC world created a 32 bit bus they call the 'local bus'. This is essentially limited to one or two slots and is used mostly for high speed video. The bus on all KiX computers is essentially our own 'local bus' but with a much better well thought out design.

The point of all this is that if you want to do high speed graphics on a 32 bit machine you need to get away from the slow AT bus. We've done this on the KiX and implemented a very high speed graphics board called the MGA or Multi Graphics Adaptor.

The MGA sits on the KiX's 32 bit bus and runs at full CPU speed. In effect it is a memory board that displays a pixel for each byte (8 bits). To display a picture you just load the video boards memory as fast as the CPU can and the MGA will display it on the screen. The MGA does this 64 bits at a time which relates to writing the full screen in 1/40th of a second or 40 times a second. This is fast, this is very very fast. We talked about doing a demo that would just blast images onto the video as fast as possible but we realized that it would be so fast that the display would just go white because of the persistence of the phosphor in the tube.

Then why make it so fast? We started out with the idea that we should make the MGA as fast as possible so that if there were delays in software or the OS at least the MGA would not contribute to it. We may have gone a bit overboard in speed but I've never heard of anyone complain about a computer being too fast.

The MGA board is 13.1 inches by 4.2 inches and has over 80 devices on it. This consists of the 64 bit video RAM, the 64 bit latch, control logic and I/O. The MGA also has a AT keyboard controller and a serial port for a serial mouse. It will work with most VGA/SVGA monitors at 640 by 480 resolution and 16+ million colors. It is full time 8 bit color. The MGA also has a ROM so that the KiX can detect if it is there and on the KiX30 how many MGAs are installed.

The MGA will be just as fast in a KiX20 as it is in a KiX30 even though the KiX20 has only one 32 bit slot and therefore can only accommodate one MGA. The MGA is supported by GWindows. As a matter of fact we designed it just to run GWindows.

The price for the MGA is \$450. GWindows is \$275. The two together is only \$599.95. On the KiX30 more than one MGA can be installed but only the first one has to have GWindows. This capability can save a lot of money.

For more information on the KiX computers contact:

==Frank Hogg==

Frank Hogg Laboratory, Inc.

# Creating Extensions to Level II

I am working on some extensions to Level II in my spare time. One that's already been released to Delphi and the OS-9 Community Network is VDGInt.ar, which includes 2 smaller versions of VDGInt. Both are MUCH smaller than the Level II upgrade VDGInt by the way! (~1800 bytes)

VDGInt\_small: No CoCo 2 graphics support. (~1600-1700 bytes)

VDGInt\_tiny : No fancy cursor positioning, either (~1200 bytes)

The appropriate choice can save you 2K of system memory! (Stock = 3300 bytes)

My current project is named pipes for Level II, which is a little bit more complicated.

Currently working:

- dir /pipe : returns a directory of all named pipes in existence
- I\$CREATE /pipe/name : creates named pipes
- read/write to named pipes: blocks if write/reads are not performed

To add:

- I\$Open /pipe/name : returns a path to the PREVIOUSLY CREATED named pipe
- del /pipe/name : delete named pipe if here are no users
- unnamed pipes : error out on write/read if read/writing process dies

The problem with named pipes lies in IOMan. You are effectively opening a path to one device, and doing IO to another. IOMan doesn't like this.

The solution: On I\$Create named pipe, create an ORPHAN path descriptor that IOMan doesn't know about. Each r/w to the named pipe uses that path descriptor, rather than the one IOMan gives it. This solves many problems, and lets data sit in a named pipe AFTER the writing process has died, and all of it's paths have been closed.

There are other problems, of course, like keeping track yourself of the number of users of the orphan path descriptor, etc. So far, I have 80% of the necessary code up and running, and the other 20% of the code will take the other 80% of the time.

Sizes: Pipeman will increase in size by NOT MORE THAN 512 bytes (2 pages). As with VDGInt, releases will be made for stock (6809) OS-9 and NitroS9 native mode

If anyone has any ideas/suggestions/wants about anything in this message, please e-mail me, and I'll see what I can do.

== Alan DeKok ==

COCO%PUCC.PRINCETON.ED



## Terry Laraway's CoCo Etcetera

Parts  
Hardware  
Hard to find items

Hitachi 6309 chip & socket	\$12
Kel Am custom 'Y' Cables	(Call)
512K Ram Chips/Kits	(Call)
2400 Baud Hayes compatible ext.modems	\$40
Serial to Parallel Interface with 64K buffer (Cable incl.)	\$50

Phone (206) 692-5374  
41 N.W. Doncee Drive, Bremerton, WA 98310

These advertisements are provided FREE as a service to our subscribers

\*\*\*\*\*

## Great Stuff

### for your OS-9 System

We've been in the software business for over 10 years--and we've developed lots of excellent software over that time. We don't have room in this space to tell you everthing, but we'd love to send you our catalogue listing all of our products. Great stuff like our *Ved* text editor, *Vprint* text formatter, *Cribbage*, *Magazine Index System*, *Ultra Label Maker*, *Vmail*, amd more.

So you only get what you need, please specify OS-9 or OS9/68000!

**Bob van der Poel Software**

PO Box 355	PO Box 57
Porthill, ID	Wynndel, BC
US 83853	Canada VOB 2N0

Phone (604)-866-5772

\*\*\*\*\*

# Q&A

**Q:** What is the "CoCoList" and how do I get it?

==Christian Miller==

**A:** It's been a while but to get on the "*coco listserver*" you need to send a message to "*listserv@pucc.princeton.edu*" with the "SUBSCRIBE" on a line in your message. You might also want to include a line with "HELP" on it. Might as well include "INDEX COCO" on another and "INDEX OS9" on another. These will get you a list of commands and available files.

==Tim Jones==

FidoNET; OS-9

**Q:** How do OS-9 Commands compare to OSK? Is OSK more cryptic?

==Aron Hsiao==

**A:** As for the commands being more cryptic, for the most part, no, they aren't. If programmers follow the Microware guidelines, then all programs should respond to `/?`, and follow assorted other conventions. Most modules' syntax is far more consistent than under Level II. On the other hand, some stuff has been ported over from Unix, so there are some real doozies. Let me give you an example:

`/dd/cmds/snd/sox`

Usage: `[-V -S] [fopts] ifile [fopts] ofile [effect [effopts]]`

fopts: `-r rate -v volume -c channels -s/-u/-U/-A -b/-w/-l/-f/-d/-D -x`

effects and effopts: various

Failed at: No input file?

This program converts sound files between different formats, and the manual really is required.

I suspect that some of the impression may come from the fact that most commands have a lot more options. Let's take dir for example:

Syntax: `dir [<opts>] {<dir names> [<opts>]}`

Function: display directory contents Options:

<code>-a</code>	show all files
<code>-d</code>	show directories with a slash
<code>-e</code>	extended dir listing
<code>-n</code>	treat dirs like files
<code>-r</code>	recursive dir listings
<code>-r=&lt;num&gt;</code>	recursive dir listing to depth <num>
<code>-s</code>	unsorted dir listing
<code>-u</code>	unformatted listing
<code>-x</code>	directory is execution dir

`-z` get list of dir names from standard input

`-z=<path>` get list of dir names from <path>

It isn't more complicated, just more powerful. OSK comes with wildcards (? and \*), so you can do things like `dir ?ons* -ar=3 -ne`. On the other hand, you can keep things as simple as you want, using just `dir`.

As for the availability of commands, well, judge for yourself. My commands directory has about 600 commands in it right now, divided over about a half dozen directories. (Yes, there's a `path=` command for OSK.) I could easily bump this up to over 1000 if I wanted to.

There are some commands that aren't available under OSK, mostly specialty ones dealing with CoCo graphics. On the other hand, there are hundreds of OSK commands that aren't available under Level II.

==Colin McKay==

FidoNET; OS-9 Echo

Origin: Micro80 Computer Club of Ottawa BBS (1:163/306)

**Q:** I don't for sure how to go about getting `gcc/g++` to you. The file is bigger than the capacity of the floppy and I know next to nothing about MS-DOS, and so have no idea how one would split a big file across two or more floppies.

==Robert Heller==

**A:** If you can get `gcc/g++` onto your OS-9 system, you have two ways to deal with your splitting problem. *Binex & Exbin* or *Uuencode & Uudecode*

I'll use Binex and Exbin in these sample:

- 1 ARCHive it first before transferring to your OS-9 system.
2. Use "BINEX" to create an ascii file that can be split by any text editor into as many files as you would like, ie. disk 1 of 9, disk 2 of 9, disk 3 of 9 etc.

The cmd line would be:

`BINEX filename new_filename <ENTER>`

at the next prompt:

Enter starting address for file: (type) `100 <ENTER>`

Enter name for header record: (type) `(real_filename) <ENTER>`

The receiver would use Exbin to un-code the file:

- 1 Join all files together (NO SPARE LINES/SPACES)
2. `EXBIN filename1 filename2 <ENTER>`  
filename1 = real\_filename that YOU gave the "header record" above.  
filename2 = what ever!
3. Check the CRC after Exbin has been used.

*Uuencode & Uudecode* are used much the same way.

==Gerry McCleary==

(FidoNET; OS9 Echo)



# Telecommunications on a Colo

This article is directed to the relative beginner in Telecommunications. Unlike many such articles, this one is geared to a specific terminal program, *SuperComm 2.2*. All of the examples given here assume *SuperComm* and a *Hayes* compatible modem.

## Setting Up:

To use a modem under OS9 it's almost mandatory to have a hardware serial port. Although there is a 'driver' that allows using the bit-banger printer port, operation is erratic and limited to 1200 bps. The stock serial port 'driver' is called *AciaPak*, and comes with Tandy OS9. Although it does work, it has problems. A tiny little buffer of 80 bytes and inability to work around certain hardware problems in the serial port itself. A replacement driver called *SACIA* is available from many online services, and solves both of these problems. Some version of *ACIA* driver must be in your boot, along with a descriptor (*/m1,/t2,/t3,etc*) for each hardware serial port you have.

In addition to the serial port stuff, *Supercomm* runs under the OS9 windowing system, making it's own window. You'll need to be in the 40 or 80 column window mode, and have */w* and at least 1 numbered descriptor in your boot that's not being used. (These things are tiny, only a few dozen bytes each. It doesn't hurt to just put all of them in).

There is a bug in the stock Tandy windowing system (imagine that!). If you haven't patched your OS9 yet, you'll have to use the command: "**supercomm<=>>>/w&**" to start it. If you have done some patching, just "**supercomm**" might work. If you get weird results when using *SuperComm*, try the first method.

*Supercomm* also stores it's autodialer files on the 'default' disk. For floppy systems, this is */d0*. The easiest way to make sure the dialer files are always available is to make a Telecom disk. To do this, put *Supercomm* itself in it's CMDS directory along with *sz* and *rz* (if you want *zmodem*). It's also a good place for *ar*, *pak*, *dearc* and other telecom related programs. On this disk, make a *SYS* directory, and inside that a *DIAL* directory. This */d0/SYS/DIAL* directory is where the autodialer files go. *Adf* files are described in detail later.

Most modern modems are 'Hayes compatible', accepting 'AT' commands to not only perform actions like dialing, but change the way the modem acts. Commonly, these changes are stored in non-volatile RAM, and herein lies a trap. If you buy a used modem, or once ran a BBS through this modem, the standard settings may have been changed from what *Supercomm* expects. If your modem doesn't seem to be acting properly, and doesn't have switches, try typing *AT&F* from the terminal screen right after starting up *Supercomm*. This resets many modems back to the original settings. If that fixes things up, use *AT&W* to make the reset permanent.

## The .adf files:

If you look at the documentation for *adf* files, there's an awful lot of stuff in there. Most of it isn't often needed, but is available if you call a service that requires different parameters (word lengths, stop bits, ANSI mode, etc.). All that's really required is this:

The first step is of course dialing out. You need to tell *Supercomm* one thing, what to send the modem.

**ADS=atdt1234567**

*ADS* (autodial string) *Hayes compatibles* use the commands **atdt(number)** for tone dialing and **atdp(number)** for pulse dialing.

When you call a service like *Delphi*, the login is pretty involved, since you are actually logging into 2 separate services: the data carrier (*TymNet*, *Sprint*) and then the BBS itself. Fortunately, *Supercomm* has quite an arsenal of **login** features:

Often, a data carrier will want you to 'blind type' something before any readable text is sent. This is what *CNS* is for- it sends as soon as the data carrier picks up.

After the first volley, login usually settles down into a question/answer format. Here is where the four search string/reply string pairs come into play. If you are used to logging in to *Delphi* manually, parts of these examples should look pretty familiar!

### TymNet:

CNS=!\*o\\*M

SS1=Please Login

RS1=delphi\M

SS2=Username

RS2=(your name)\M

SS3=Password

RS3=(your Password)\M

### SprintNet:

CNS=!\*@!\*D!\*M!\*M

SS1=@

RS1=c delphi\M

SS2=Username

RS2=(your name)\M

SS3=Password

RS3=(your password)\M

The *\\** is a half second pause. Since my service is a little slow, I've got lots of pauses in there! The *\M* is an <enter>. (Must be a capital M)

One last thing that's nice to have is a set of canned keyboard macros. These are labeled *KM1* through *KM8*, and are sent when *ALT-(number)* is pressed while online.

*KM1*=go top\M

*KM2*=com os\M

*KM3*=com coco\M

*KM4*=mail\Mselect\Mextract /all grab.txt\Mdel /all\M

These are all *delphi* commands. Note how I've crammed quite a few of them into one macro- this works with *Delphi*'s type ahead buffer. Some other BBS might require a few *\\** pauses after each <enter>.

## Normal Telecom and Conference:

It's easy enough to navigate through most services using the



menus, but sometimes the people you meet there seem to avoid English at all costs! These are the most common abbreviations:

-:) A Smiley Face (turn your monitor on it's side if you don't see it). There are lots of variations, from winks ;-)  
to looks of amazement :-O to frowns :-(  
**BTW** By the way  
**BRB** Be right back  
**B** Back  
**IMHO** In my humble opinion

Conferences can get a bit confusing, since text comes in at any time, clobbering the line you're typing. *Supercomm* has a 'conference mode' (ALT-z). In this mode you get your own personal window at the bottom of the screen to type in.

There is one more way to send text, and that is by redirection. Clear to a free window with a shell in it, and issue any OS9 command that writes to the screen (even list). But add redirection to the port *Supercomm* is using. When you see somebodies module directory fly by on screen, this is what they've done: mdir >/t2

If you are using the stock ACIAPak driver, OS9 sends stuff too fast for some services to handle, and characters get lost. There is a program called *slowio* in the databases that slows things down enough to avoid this problem, but the best solution is to use SACIA.

#### File Transfers:

*Supercomms* file transfer menu is called with **Shift Up Arrow** for upload or **Shift Down Arrow** for Download. The file transfer protocols are described below:

**ASCII transfer** - or 'buffer capture' This method simply grabs anything going to your screen, or from your keyboard, and puts it in a buffer in RAM. When the buffer fills up, it's saved to a disk file. Buffer captures are mainly for text only, since there is no error correction involved.

So you can see there are 2 things involved here the buffer in RAM and the file on disk. *Supercomm* gives you control over both in a variety of ways.

When you start *Supercomm*, one of the parameters you can give it is a buffer size. The default is a small 2K. It fills up fast so it writes to the disk often, which slows things down, especially with Tandy disk controllers. But, it's very safe. If your machine crashes the most you can lose is 2K. It's up to you to judge how much safety to trade for speed. This buffer doesn't have to be capturing all the time. The **Alt-m** key opens and closes it so you can select just the parts you want saved.

The second part is the disk file. It can also be specified on the command line, in two different ways. **f=pathlist** starts up with the buffer capturing. **-f=pathlist** starts up with the buffer closed, but ready to go as soon as you hit Alt-m. If you don't set up a capture file when you start *Supercomm*, you can always start one from the download menu, just shift downarrow and select ASCII.

*Supercomm* will normally leave this file open until you quit for the day, so it will always be ready to accept more text. But there are times when it should be closed. You might want to change the file you are capturing to, or perhaps you have everything you want to capture, and just don't want to leave the file open any longer. Just use the download menu again, and *Supercomm* will ask if you want to close the file.

It's possible to do the reverse of a buffer capture. (I leave you to figure out what that might be called!). If you select ASCII from the upload menu, SC will ask for a filename, and send it up just like you had typed it online. This is a handy way to post messages. They can be typed up with any text editor and saved on disk. Then, while online, fill out the address part, and at the 'Enter your message' prompt, upload the file! There are ways to include the address in the ASCII file, but exactly how depends on the service.

The rest of the choices are error correcting 'protocols', good for sending and receiving programs and 'sensitive' text like source code. To use any of them, you start the other guy downloading or uploading, then use shift-arrow to start your side. (Except

for Zmodem- see below)

**XModem**- is the oldest (and slowest) of the error correcting transfers *Supercomm* offers. It sends tiny little 128 byte blocks, checking for errors between each. Handy if the service involved doesn't know any better, or if the phone lines are really bad. XModem needs to know the filename for the download- Be careful to use the same extension as the service gives you, since many files are archived. Later on, you will need to know if it's an .ar, .pak, .arc, or whatever type file. *Supercomm* attempts to help out here, grabbing the name you typed and popping it up in the filename window. You can either use it, or enter a different name.

**XModem 1K** is the same thing as XModem, but with larger chunks between the error checks. It still needs to have the filenames typed in by hand, but it sends 8 times as much data before checking errors. Acts just like above. Many people call this YModem, but it's not!

**YModem (batch)** YModem adds 2 features to XModem 1K. First, it can get or send the filenames, so you only have to type them once, to the sender. (That's them for downloads or *Supercomm* for uploads). The second feature is, it can handle more than one file at a time. On services that support 'groups' (like Delphi), just give them a DOWN /ALL, call up YModem, and go get some coffee! Note that batch isn't really part of the name, YModem is always a batch protocol. It's just so many people call XModem-1K YModem....

*\*Note- Supercomm will make an effort to reformat ASCII files for you. The options menu (Alt-o) has an Auto-ASCII selection that will strip or add linefeeds and such for compatibility with MSDOS type machines. Turn Auto-ASCII On and if it works properly on your service, use update SC (alt-u) to make this a permanent feature*

**ZModem** isn't built into *Supercomm* like the other transfer methods. The two files sz and rz must be available in the CMDS directory with *Supercomm*.

ZModem is the latest and greatest transfer protocol supported. It combines



# Club Activities Report

*Bellingham OS9 Users Group - Longview/Kelso CoCo Club  
Mt. Rainier CoCo Club - Port O'CoCo Club - Seattle 68xxx Mug*

## Bellingham OS-9 Users Forum

We canceled our October meeting due to conflicts of everyone involved. The September meeting was not posted in the *Newsletter* last month due to a lack of space. So now that space is available, here are our September meeting minutes:

**Rodger** contacted GIMIX via telephone and found some answers to the disk driver/descriptor problems that we have been wrestling with. Turns out that GIMIX did not follow OS-9 convention. Instead of placing the Hard Drive data in the descriptor (# of tracks, # of heads, etc.), they placed a descriptor table in the driver module. The descriptor simply points to the appropriate table in the driver module. There are not official documents available. GIMIX no longer "officially" support their older 6809 machine. So **Mike Pleas** was given the assignment of decoding the GIMIX assembly code for the Hard Drive driver module in order to determine the location of the proper table for the hard drives we have available.

We are still very pleased with ourselves over the successful completion of the "Parallel PIA Port" for the CoCo-3 and the "MFM-RLL Hard Drive Interface" card". Mike gets most of the credit for solving the problems with the Interface. **Scott Honaker** of the Seattle 68xxxMUG is researching information for us so that we can modify the parallel port to be *bi-directional*.

Our other project at the September meeting was to re-wire at least one of the terminal cables for printer usage. The handshaking line needed to be enabled. **Wes Payne** managed to find the appropriate 25 pin RS-232 pin out diagrams to compare the terminal and printer differences. Whole project was completed in a half hour.

==Rodger and Barbara==

## Port O-CoCo

October's meeting is the beginning of our new series of tutorials. We started promptly at 7p.m. with a few announcements and passing around of samples of various publications **Terry Laraway** has received. Since we had some new faces and several returnees from long past meetings, we took time to introduce ourselves and mention our interest areas and areas of strength.

Just to break the ice with our tutorial topics **Donald Zimmerman** explained just where we got the world BASIC for the computer language. It stands for **B**eginners **A**ll-purpose **S**ymbolic **I**nstructional **C**ode. For that matter, this crazy naming system is old hat. One of the first computer languages is COBOL. In case you didn't know, COBOL stands for **C**ommon **O**rdinary **B**usiness **O**riented **L**anguage written by Grace Hopper while in the Navy. She also originated use of the word "bug" when something goes wrong with a program. If you want the full story on that, we can stick it in for a Holiday Treat.

**Gene Elliott** began our formal presentation by showing a video tape of the output of his CoCo while running several short, simple programs. He underscored the sophistication of our "dialect" of BASIC (by Microsoft) and how much more powerful it is than the BASIC available for many other machines. Also our BASIC is part of each and every machine (in a special ROM chip). Almost all other machines have to have BASIC installed, which means in some cases you will have to purchase BASIC as an application before you can do any programming in BASIC.

Gene wound up this installment of his presentation by handing out a

program in BASIC for a perpetual calendar. The challenge is that there is ONE error in the program; not enough to crash the program, but just to give you incorrect information. Our challenge is to find the one error and correct it. Will anyone take the challenge? Stay tuned!

For those who didn't attend the meeting, Gene has extra handouts. Just send him a long SASE to Gene Elliott, 6905 Corfu NE, Bremerton, WA 98310.

After a brief break we started on the OS-9 portion of the tutorial, spear headed by **Mark Kulien** and **Chris Johnson**. Since we had two systems set up and running, the group broke into two and gravitated to the topic that helped them the most. Both Mark and Chris talked about setting up your system. It was obvious that they were clearing out confusion and frustration because both of them worked with members of the group past 10p.m.

We are continuing on this BASIC and OS-9 tutorial at both our November 15th and December 20th meetings. All are welcome. Don't feel that you can not benefit from these fine presentations because you missed the first meeting. The presenters, as is the club as a whole, are anxious to assist as many people as possible. Gene will be continuing the BASIC presentation and Mark and Chris will be penetrating the clouds of mystery surrounding OS-9.

Port O' CoCo was asked to create small handouts of the various computer clubs in Kitsap County. We have been doing this for a couple years. This week we provided 1,000 listings to Kitsap Greater Service to give to new people as they move into the country. The listing also promotes the PNW CoCo FEST IV in June of 1994.

REMINDER: There is a Computer Swap Meet at the Kitsap Pavilion

November 13th. We have rented space there to promote the club and the CoCo. We look forward to seeing many of you there!

==Donald Zimmerman==

**FIND THE BUG CONTEST**

**Request your BASIC listing with the hidden bug ASAP!**

**Just send a SASE to:  
Gene Elliot  
6905 Corfu NE  
Bremerton, WA 98310**

Sponsored by Port O'CoCo Club

**Seattle 66xxxMug**

Rodger Alexander began the October meeting by showing off his working prototype hard drive interface card. Schematic diagrams were made available to everyone at the meeting. After demonstrating that the computer actually booted up and the read/write functioned correctly, Rodger pulled the board out of the computer (tower case) and passed it around for examination. The board uses "point to point" wiring with wire wrap type (30 gauge) wire and has exactly the same dimensions as the Burke & Burke CoCo XT Hard Drive Interface Card so that it can fit into the Burke & Burke Interface case.

Scott Honaker was going to demonstrate Packet Radio (Ham Radio BBS) but unfortunately did not have the proper software on the computer. His presentation included a very small TNC terminal unit that included its own software so it will run on any type computer that has a terminal program and an RS-232 port. It also has sufficient RAM so that it can store messages while the computer is off. A "mail" indicator light lets you know that it is holding mail for you to read when it is convenient to turn on the computer. Very Cool! The unit cost \$110 - \$120. The new No-Code Amateur License makes it very easy for anyone to get a HAM License with only a couple of hours of studying FCC Rules, Frequency

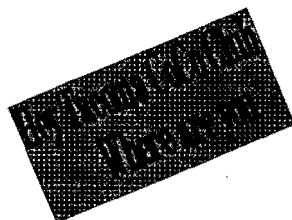
Allocations and Amateur Procedures. Radio Shack sells a "New No-Code Technician Class FCC License Preparation" book for \$9.95

Scott also brought along a 2-meter receiver and a 2-meter transmitter circuit board designed to operate together as a transceiver. Both circuit boards, less crystals, are advertised for sale in the *QST Magazine* for \$40. Scott plans to use the units to set up an Amateur BBS to handle Voicc Mail. It will be the only one in the country and will take advantage of the new audio compression software available on the latest *MS Windows* updates that is capable of compressing one hour's worth of audio onto 4 megs of hard drive space. *WOW!*

News about the cancellation of *The Rocket* by Burke & Burke lead to a heated discussion. Comments ranged from disappointment to criticism and to possible rationale of the failure. Several voiced a concern that it would mean the certain death of the CoCo. It was also suggested that Chris Burke may make the schematic and support software available. Those who would be willing to build *The Rocket* would still have to purchase the Microware OSK "Industrial" Package.

Next month's meeting will feature a beginner's demonstration on how to build an OS-9 Level Two Boot Disk on a CoCo-3. Also, we hope to get Mark Kulien to make a presentation on OS-9 Software.

==Barbara Alexander==



**CoCo Hardware**

- ST-225 20Meg Hard Drive \$50
- Deluxe RS-232 PAK \$25
- PBJ 6-Slot MultiPak \$50

Call 1-206-734-5806

Prices do not include shipping cost

**Telecommunications**

*Continued from page 9*

all of the features given above. The block size changes, getting smaller if the phone lines are bad, and larger if they are good. Grabs the names and does multiple files like YModem. And *Supercomm* supports another feature Auto-Z! If the host starts sending a ZModem transfer, or is waiting for you to send one, ZModem pops up on it's own. To get a file from Delphi, you type DOWN, and thats it- no menus or filenames needed!

If you have rz and sz, go to the option menu (alt-o) and turn Auto-Z on, then update (alt-u) to make it permanent. Thats about it! Feel free to contact me with any questions or comments on this file, and I'll put them in the update pile!

==Rick Uland==

InterNet : rickuland@delphi.com

**CoCoPRO Software  
STILL AVAILABLE**

- Data Windows* \$39.95
- Multi-Menu* \$14.95
- together: \$15 + \$2 S/H
- Newspaper 09* \$34.95
- News Fonts* \$ 9.95
- The Zapper* \$14.95
- together: \$15 + \$2 S/H
- OS9 Level II BBS* \$19.95
- Disk Manager Tree* \$18.95
- Tools II* \$24.95
- together: \$15 + \$2 S/H
- Level II Tools* \$19.95
- Presto-Partner* \$19.95
- Data Merger* \$12.95
- together: \$15 + \$2 S/H

SUPER DEAL--ENTIRE PACKAGE  
\$55 + \$5 S/H

Rick's Computer Enterprises  
P.O. Box 276  
Liberty Kentucky 42539

**FREE Classifieds  
for subscribe**

## Washington State BBS List

### **COLUMBIA HTS. BBS**

-- Longview/Kelso --

RiBBS (FidoNET)

(206) 425-5804

### **DATA WAREHOUSE BBS**

-- Spokane --

RiBBS (FidoNET)

(509) 325-6787

### **BARBEQUED RIBBS**

-- Bellingham --

PC-Board (PC-Net) - CoCo Conference #5

(206) 676-5787

### **●S-9 TACOMA BBS**

-- Tacoma --

RiBBS (FidoNET)

(206) 566-8857

### **ULTIMATE EXPERIENCE BBS**

-- Anacortes --

RiBBS (MaxNET)

(206) 299-0491

## ***Bellingham OS-9 Users Forum***

**OS-9 and the Color Computer \$7**

*Tutorial and Hardware Hacker's Manual.*

Includes 5-1/4 Disk of (360K) of upgrade software

**Color Computer Video Library \$10**

Fixing the MultiPak IRQ \* Installing Floppy Drives

Installing 512K Memory \* Installing B&B Hard Drive

**OS-9 Newsletter \$12/yr.**

12 monthly issues packed with OS9 Update, Tutorials,

Listings, Classifieds and PNW "Club Activity Reports"

Subscriber's Technical Support (206) 734-5806

Mail your order to: ***Bellingham OS-9 Users Forum***

***3404 Illinois Lane, Bellingham WA 98226***

### **COPYRIGHT NOTICE**

The *OS-9 Newsletter* is a copyrighted publication by the Bellingham OS-9 Users Forum; Rodger Alexander, Editor. Duplication and/or distribution is prohibited without written permission of the editor.

***OS-9 Newsletter***

***3404 Illinois Lane***

***Bellingham, WA 98226-4238***