

OS-9 Newsletter

Volume III No. 2

Bellingham OS-9 Users Group

February 29, 1992

Baisc09 Part-5

Tutorial

by Scott Honaker & Rodger Alexander

Before we write our SEARCH, SORT and PACK procedures for our database program, it would be helpful if we see how *Basic09* reads data statements and searches data to find desired or chosen data variables, such as the address of a name entered in the database.

When developing a SEARCH Procedure, a great deal of thought needs to be given to the method or process used to find and match strings. The most obvious and easiest to construct would simply be a search of all of the variables in all of the records from the first record to the last record at the end of the file. In a small database, this would be satisfactory. However, as the database becomes larger and larger, the amount of data stored becomes massive and the time to find data located near the end of the file could take a great deal of time.

One solution is to divide and conquer using the "binary search method". In this method, the database "data" is divided in half and a comparison is made to see if the requested record is located in the lower half or the upper half. The half of the data in which the requested record is determined to be in is divided again. Now a comparison is made to see which 1/2 of the original half the requested record is in. This process is continued until the search is narrowed down to two records. If the first record doesn't match the search criteria, then obviously the second record is the requested data.

You might want to read the above paragraph a few times to make sure you understand it. It seems simple enough, but when you try to follow the process in the actual program it gets pretty weird!

Type in the listing below and run it. After you get all the bugs and errors out of it, see if you can follow the logic of the search pattern. By entering the name of a CITY you will get the name of one of our OS9 Newsletter Subscriber.

```
PROCEDURE jump_search
DIM lname(15):STRING; city(15):STRING;
choice:STRING
N=15
FOR I=1 TO N
READ lname(I),city(I)
NEXT I
10 E=N
B=1
```

-- IN THIS ISSUE --

Jump Search Basic 09 Tutorial	Pg 1
Bernie's Bit Bucket Call Memo Basic09 Program	Pg 2
Tower Fest -2 More CoCo's into PC type cases	Pg 4
Classifieds Software/Hardware FOR SALE	Pg 5
Keyboard Extension Cable Make your own extension cable	Pg 6
Getting Started with Multivue Making a new Multivue Boot File	Pg 7
Tips from Terry Laraway 512K Ram - IC Relay - Replace Disk Rom	Pg 8
OS-9 Concepts MultiPak Interface and I/O Devices	Pg 8
Club Activity Report 68xxxMUG - Longview - Port Orchard-Bellingham	Pg 10

```
P=0
PRINT "Enter City:";
INPUT choice
20 D=INT((E+1-B)/2)+B
IF E-B<3 THEN 50
IF choice<>lname(D) THEN 30
P=D
GOTO 70
30 IF choice<lname(D) THEN 40
B=D+1
GOTO 20
40 E=D-1
GOTO 20
50 FOR I=B TO E
IF choice<>lname(I) THEN 60
P=I
60 NEXT I
70 IF P<>0 THEN 80
PRINT "DATA NOT FOUND>"
GOTO 10
80 PRINT lname(P),city(P)
GOTO 10
DATA "ANACORTES","DAVID FISHER"
DATA "AUBURN","GARY BONDAHL"
DATA "BELLINGHAM","CRAIG DUBOIS"
DATA "BREMERTON","TOM BROOKS"
DATA "CONCRETE","SKIP MC_NEALEY"
```

```
DATA "FAGLE_ID", "DON STRUNK"
DATA "KETCHIKAN", "BERNIE BESHESER"
DATA "LACEY", "DON ANDERSON"
DATA "LONGVIEW", "STEVE HAMMOND"
DATA "POULSBO", "BOB FOSTER"
DATA "PORT_ORCHARD", "BOB VAIL"
DATA "RICHLAND", "ALAN ROCKWOOD"
DATA "SEATTLE", "ROBERT BIELKA"
DATA "TACOMA", "ALAN JOHNSON"
DATA "WHITLAW_MI", "KERRY KOWALSKI"
```

To really understand how the "binary search method" works, select one of the cities.....like.....LONGVIEW. Looking at the data list in the Procedure, Longview is the ninth city in the list, OR--- "Iname(9)". Knowing which variable number we are looking for permits us to follow the programs logic in it's search to find "LONGVIEW".

Start at line 20 and calculate the variable arguments. In the next line, the value of "E - B" (15 - 1) is greater than 3 so we continue to the next line. "D" = 8, therefore "Iname(8)" does not equal LONGVIEW ("Iname(9)") so we go to line 30.

Line 30 wants to know if our selection, LONGVIEW ("Iname(9)") is less than "Iname(8)"? The answer is "NO" so we go to the next line in which "B" becomes equal to "D + 1" (8 + 1 = 9). Now we return to line 20 and start over with a new value for "B".

Line 20 now becomes "D = INT((15 + 1 - 9)/2)+9" "D" now equals 12. In the next line "E - B" translates to 15-12 which is equal to 3 which is still not less than 3 so we continue on to the next line. LONGVIEW ("Iname(9)") certainly does not equal "Iname(12)" so we jump to 30 again.

Line 30 ask if LONGVIEW ("Iname(9)") is less than "Iname(12)". YES!!!! So we jump to line 40.

In Line 40, "E" becomes 12-1, or "11". Jump back to 20. This time "D = INT((11 + 1 - 9)/2)+9". "D" = 10. Now the next line is true. "E - B" is less than 3 (11-9 = 2), so we jump to line 50.

Line 50: "FOR I = 9 to 11". The next line does a comparison again to see if LONGVIEW is NOT equal to "Iname(9)". But at last, "choice" does equal "Iname(1)", so the variable "P" becomes equal to 9. When the FOR/NEXT loop expires we are finally jumped to line 80 and the found record is printed to the screen.

Next month we will apply this method to our *Database*
-- Rodger Alexander --

Special thanks to Bernie Bersherse for the "Jump Search" article by Jerry Sturdivant in the September '85 *COMPUTE!*. It was very helpful.



Bernie's

"BIT_BUCKET"

Callmemo is a short Basic09 program that can turn those miles of only half used sheets of pin-feed paper into useful, ornamental, decorative, recycled telephone memorandum forms. If you are like me, you spend simply THOUSANDS of dollars each year on these forms at the local office supply store each year to keep your wife from having to take your messages from V.P. Dan Quayle on scrap paper. Well, here comes Basic09 to the rescue!

The first thing to tell you is that I use a Tandy DMP-130a printer so that's how the printer codes are set. If you want to change the printer codes, they are found at approximately:

```
050D    set printer to BOLD
0543    set printer to Double-Wide
055C    cancel Double-Wide
068B    set Double-Wide
06C4    cancel Double-Wide
```

I say "approximately" because if you wish to substitute, "While you were out fishing with your brother, George" for "CALL MEMO" then the offsets will be different. For those new to Basic09, what looks like line numbers here are actually offsets from the start of the program.

NOTE: Don't use a vertical line character in the middle of your double-wide line.

The technique I used to type in *CALLMEMO* was to use my favorite word processor, Bob van der Poel's *VED*. I typed in the lines, NOT using a space to begin a line and with word wrap disabled. When I got to the place where I defined the long strings (like star:=, undlin:=, etc.) I hit a carriage return after the "equal" sign, (=) and after I had all the lines adjusted to my satisfaction on the screen, simply went back through and deleted the excess carriage returns! To do it is easier than talking about it.

Another reason I like to program with *VED* instead of the built in Basic09 editor is the ability to block copy. That came in handy in the lower part of the program where there are so many tedious, repetitious, monotonous, tiring lines to type in. Another benefit of using block copy is that you make exactly the same mistakes in the second half. You don't waste time looking for new mistakes! Incidentally, Kevin Darling's patch to the SCF module that allows improved editing of the command line works in Basic09, also. Get it. It makes life easier, too.

Of course, Basic09 will let you know when and where you have made any typos when you start to load the finished product into Basic09 to run it. Don't lose heart! That is why we use OS-9 with windows! Just work on the source file with *VED* in one window while looking at the error messages on Basic09 in another. With *VED* I use the "save and keep working" command and then kill the program out of Basic09 and reload the new source. By working in the same directory you can do this without even breaking a sweat.

A note about how Basic09 reports mistakes is in order... If the line with the reported error looks good to you, then take a close look at the line just ahead of it. Make sure you have your variables spelled correctly, etc.

I deliberately didn't write a fancy ending on this program, asking if you want to make some more call memos, etc. I figured that I could just punch the button on

my mouse again if I had more paper to recycle. If you run most things from a hard drive and *Multi-Vue*, like I do, then it's almost as fast as keeping the program in memory, and less hassle than having to interact a half dozen more times just to get the bloody program to die! If you are running from a 128K CoCo with 156K floppies then you'll appreciate the savings in memory and disk space.

LISTING 1:

```

PROCEDURE callmemo
REM CALL MEMO GENERATOR
REM printer oodes for TANDY DMP-130a
REM use the back sides of used, pin-feed paper
DIM blank:STRING[80]
DIM callmem:STRING[80]
DIM star:STRING[80]
DIM callto:STRING[80]
DIM clfrom:STRING[80]
DIM undlin:STRING[80]
DIM telnum:STRING[80]
DIM urgent:STRING[80]
DIM plzcal:STRING[80]
DIM retcal:STRING[80]
DIM recdby:S RING[80]
DIM printer:BYTE
DIM count,number:INTEGER
blank:="          P          |"
star:="*****|*****"
callmem:="          CALL MEMO          CALL MEMO"
callto:="TO:          |          TO:"
clfrom:="FROM:        |          FROM:"
undlin:="          |          "
telnum:="TEL # (    ) - EXT          | TEL # (    ) - EXT"
urgent:="[] URGENT          [] CALLED-NO MESSAGE | [] URGENT          [] CALLED-NO MESSAGE"
plzcal:="[] PLEASE CALL    [] WILL CALL LATER   | [] PLEASE CALL    [] WILL CALL LATER"
retcal:="[] RETURNED CALL  [] WAS HERE         | []RETURNED CALL  [] WAS HERE"
recdby:="REC'D BY:        DATE:          TIME:   | REC'D BY:        DATE:          TIME:"
PRIN CHR$(12)
PRINT \ PRINT \ PRIN \ PRINT
PRINT USING "s80^", "CALL MEMORANDUM GENERATOR"
PRINT \ PRINT \ PRINT
PRINT USING "s80^", "How many sheets of paper do you want turned into call memos?"
RUN gfx2("curxy",39,13)
INPUT number
PRINT \ PRINT "          O.K., I guess its time for me to go to work!"
PRINT \ PRINT "  counting . . . ."
PRINT
REM Printing begins here
OPEN #printer,"/p":WRITE
PRIN #printer,CHR$(27); CHR$(31);
FOR count=1 TO number
PRINT USING "i4",count;
PRINT #printer,blank
PRINT #printer,CHR$(27); CHR$(14);
PRINT #printer,callmem
PRINT #printer,CHR$(27); CHR$(15);
PRINT #printer,blank
PRINT #printer,star
PRINT #printer,callto
PRINT #printer,undlin
PRINT #printer,clfrom
PRINT #printer,undlin
PRIN #printer,telnum
PRINT #printer,star
PRINT #printer,urgent
PRINT #printer,plzcal
PRIN #print

```

LISTING 2:

```

PROCEDURE beep
PRINT CHR$(7)
END

```

Two parting shots:

1. Turn the cut pad of the call memos face down to staple them together. This allows them to be torn off the top much more cleanly.
2. If someone actually gets, "While you were out fishing with your brother, George" to print in double-wide, please write me a letter and explain how you did it.

Bernie's Bit Bucket
P.O.Box 9381
Ketchikan, Alaska 99901
1(907)225-1324

□ Tower Fest - 2

Well, we did it again. Sticking Color Computers into PC type cases. This time under the sponsorship of the Seattle 68xxxMUG. Lot of the same faces too. Those in attendance were Robert Bielka, Jeff Brittan, Scott Honaker, Phil Johnson, Randy Kirschenmann, Terry Laraway, John Schliep, Scott Worley, Donald Zimmerman and Donald Zongker.

This session was quite a bit different than the first one at Port Orchard several months back. A couple of the guys already had the CoCo's in cases from the previous session, but were there for technical support as they modified their installation and/or added more peripherals to their computer.

John Schliep has everything working on his system, but has yet to find a satisfactory mounting so that everything fits. His power supply is mounted on the bottom of his case rather than the top which is typical with most tower cases. His power supply is also physically longer than others and simply messes up every possible location of the computer and the multipak. Finally we arrived on the only solution, which was to mount the mother board along the top, facing down, and the multipak board vertically along the side. Hey, it works! John also had some problems with cold solder joints to his power connections. Flowing new solder over the old joints solved that problem.

Donald Zimmerman was having trouble with his 3-1/2 inch drive so we (actually Scott Honaker and Scott Worley) determined that his Radio Shack disk cables were not compatible with his 720K drive so they put on a new standard card edge connector to replace the Radio Shack one (with the missing pins), and everything worked. A little later however, the computer completely crashed with garbage on

the screen. Oh, Oh, a bad 6809 chip. Nope, just a loose GIMME chip. A little push down on the chip in it's socket revived the CoCo. We also messed around with Donald's OS-9 setup. Here's a question for everyone: **Why does Donald's hard drive system refuse to read the default drive /DD (/HD) during boot up?** (I hope Chris Johnson knows the answer).

Robert Bielka had a large standard AT type case. His power supply was not standard, but small. After asking a few questions, getting his "Y-Cable" from Terry and his instruction booklet, Bob took off for home, confident he would have no problems completing the task.

Phil Johnson presented the most difficult problem because of his 1Meg memory board and his 512K Ram Disk Card which is a good 1-inch longer than most I/O boards. His 1Meg memory board sticks up too high to permit a neat installation of the mother board under the disk drives, and of course he had the old Multipak which required major "hacking" (with a hack saw) to make it fit. Phil's horizontal AT case was 1/8th of an inch shorter than the width of the Color Computer mother board forcing us to install it on the bottom of the case and the now much smaller multipak along the back side of the case. That required an 8 inch length of "Y-cable" which is always a concern. Phil finally went home around 10 p.m. with everything installed but not tested.

Jeff Brittan on the other hand, had the same horizontal AT case that Phil had but no high stacked memory board problems, so his motherboard fit neatly under his disk drives and his multipak (also hacked sawed) along the side of the case, giving him plenty of room to plug in whatever I/O boards he wants.

Donald Zongker was there to make minor adjustments to his installation and saved everyone else's installation with a supply of

every conceivable odd part that was needed. Would you believe a 1 foot length of ribbon cable with a 10 pin header at one end (for the monitor jack on the computer) and a DB-9 chassi mount jack on the other end for 50 cents! !!! However, Donald did have some problems of his own, losing power to his multipak and trouble shooting some broken jumper wires.

We started at 10 a.m. at Jeff Brittan's restaurant supply shop, and finished after midnight. It was lots of fun. I especially enjoyed the hack saw modifications to the older multipaks. Now that I was successful with theirs, I feel confident to hack mine! I filled up with coffee and drove back to Bellingham. Only fell asleep once, but bounced out of that ditch at 70 mph without a scratch!

Something new I learned. The larger AT power supplies have a "Power Good" test line (Orange wire on one of the six pin jacks). This line expects to find 5 Volts D.C. being returned to it from the mother board. Also, the Power Supply will not even turn on unless there is a load across it's output. Both conditions must be met before the power supply will work.

If you are planning to put your CoCo into a PC case, a couple of suggestions. Most important, take a look at a couple of installations before you go out and buy your PC Case so you can see what kind of space requirements are needed and what kind of configurations are possible. Make sure that the case includes a power supply. Mini and Midi towers and full size AT cases seem to work best. And finally, I have prepared a 10 page set of instructions with pictorial examples which I will gladly send anyone wishing a copy for 79 cents.

Rodger Alexander

Classifieds

HARDWARE:

CoCo3 - with 512K	\$ 90
F-502 with 2nd Single Sided Drive	\$110
Magnavox Color Monitor #40	\$ 60
DMP 105 Printer	\$ 60
MOD CCR81 Tape Recorder	\$ 13
Deluxe Joystick	\$ 5
Regular Joysticks (2)	\$ 4
DC Modem Pak	\$ 5

SOFTWARE: (Complete with Documentation)

Home Publisher	\$ 7
Cash Budget Manager	\$ 7
Telewriter-64	\$ 14
Telewriter-128	\$ 16
VIP CALC	\$ 14
TS Spell/Edit/Word	\$ 15
Investrograph	\$ 9
Hi Finance	\$ 5
Disk EDTASM	\$ 13
Master Dir.	\$ 4
Start OS-9	\$ 13
Basic09	\$ 11
MultiVue	\$ 11
Zone Runner	\$ 5
Rescue on Fractalus	\$ 5
Rogue	\$ 5
Sub Battle Simulator	\$ 7
Ghana Bwana	\$ 7
Aces	\$ 4
Pro Football	\$ 4
Classic Solitaire	\$ 5
La Belle Lucie	\$ 5
Chess	\$ 4
Checkers	\$ 4
King Quest III	\$ 17
Barbarian Quest	\$ 5
War Game Designer	\$ 4
Koronis Rift	\$ 5
Ironsides & Red Sails	\$ 6
Interbank Incident	\$ 6
Flight Simulator II	\$ 6
Cook Book	\$ 7
Planet Engine	\$ 5
CoCo Max III	\$ 20
Rainbow Indexes	\$ 5
Weekly Winners	\$ 4
Super Disk	\$ 4
CC3 Flags	\$ 4
Puzzles	\$ 4
Geography Quiz	\$ 4
Tetra	\$ 4
Scrabble	\$ 4
Pictures (4-Disks)	\$ 20
Carmen Santiago	\$ 11
200 Misc. Programs on 16 Disk	\$ 8

DL Logo	\$ 11
Typing Tutor (tape)	\$ 5
Color Stats (tape)	\$ 5
Mickeys Write (tape)	\$ 5
Telewriter-64 (tape)	\$ 10
Pyramid (tape)	\$ 5
1985 Best of CoCo	\$ 4
Color File	\$ 5

ROMPAKS:

Bridge Tutor	\$ 11
Spectaculator	\$ 5
Color Scriptsit	\$ 5
Dungeons	\$ 5
EDTASM	\$ 5
Thexder	\$ 5
Sokoban	\$ 5
Slipspeed	\$ 5
Castle Tgaroggao	\$ 5
Tetris	\$ 7
Robo Cop	\$ 5
Color Baseball	\$ 5
Shanghai	\$ 5
Arkanoid	\$ 7
Super Pitfall	\$ 5
Springster	\$ 5

COCO BOOKS:

Rainbow Guide to OS9	\$ 12
Basic09 Tour Guide	\$ 12
CoCo Assembly Language Programing	\$ 10
CoCo Graphics	\$ 10

Back Issues of Rainbow from April 1985 2 each

Contact: Roland Kennedy, 1003 Shalimar Dr.
High Point, NC 27262
(919) 887-2236

BENEFITS TO SUBSCRIPTION MEMBERS

As a subscribing member of the Bellingham OS-9 Users Group, you enjoy the following benefits:

1. Monthly Newsletter
2. FREE Public Domain Library (6 Megs)
3. FREE Classified Ads
4. Technical Assistance (734-5806)
5. On-Line OS-9 Conference (676-5787)

SUBSCRIPTION INFORMATION

The OS-9 Newsletter is compiled and printed monthly by the Bellingham OS-9 Users Group. Subscription rates are \$5 for 6 months or \$10 for 12 months. Mail your subscription check to: Rodger Alexander, 3404 Illionois Lane, Bellingham, WA 98226

CoCo Keyboard Extension

How many times have you wished that your computer desk could be neater instead of a bunch of cables, multipak, more cables, disk drives, more cables, etc. If you could just hide the CoCo out of site somewhere with only the keyboard, disk drives and monitor on the desk. It would be so neat.

I've seen those advertisements for keyboard extender cables, but I'm just too cheap to spend the money. Besides I like to "hack" around a little bit, and it couldn't be that hard to make some sort of keyboard extension, right?!

There are several approaches to making up a keyboard extension cable, but the simplest (no soldering required) is presented here for your consideration.

Parts:

- 2- 34 Conductor *Kel-AM* Card Edge Connectors
- 4 to 8 ft. of 34 Conductor Ribbon Cable
- 1 Discarded ROM Pak
- 11/16th x 1 inch piece of tag-board or thin card-board

*NOTE: *Kel-AM* Connectors are available from Terry Laraway, (206) 692-5374 for approximately \$6 each. Standard 34 Card Edge Connectors reverse pin order and will not work. You must use the *Kel-AM* Connectors for this project.

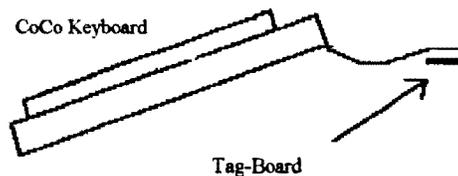
Construction:

The idea is simple. Take a desired length of cable and devise a way to connect the keyboard at one end and attach the cable to the computer at the other end. The *Kel-AM* Connectors simplify the project by eliminating the need to do any soldering.

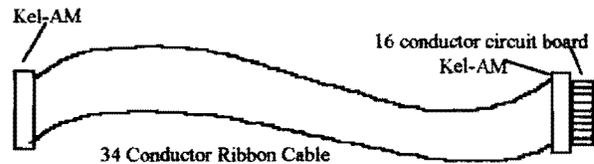
1. Press on the *Kel-AM* connectors (use a vice) to each end of the desired length of 34 conductor ribbon cable.
2. Remove the circuit board from a discarded ROM Pak and cut off the part of the circuit board that plugs into the computer, using a hack saw. You only need 16 conductors on one side of the circuit board section that you cut off from the ROM Pak, so trim the length of the section to 11/16ths. You should be able to plug the circuit board section into one of the *Kel-AM* Connectors with all 16 conductors lining up with the 16 contacts in the connector. The overall dimensions should be 11/16ths by 1/2 inch.



3. Use a file to reduce the thickness of the circuit board section without damaging the 16 conductors on one side. Be careful to file only a 1/4 inch wide strip along one edge of the circuit board to a thickness that will permit the circuit board to insert easily into the keyboard jack on the Color Computer. The other edge of the circuit board will plug into one of the *Kel-AM* Connectors.



4. Use *Elmer's Glue* to attach a piece of tag-board or thin card-board (such as used to package electronic components) under the mylar keyboard cable so that the combined thickness of the mylar cable and the tag-board will provide a solid contact when inserted into the other *Kel-AM* Connector.

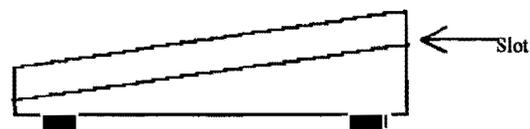


Testing:

With the 16 conductor circuit board plugged into one end of the cable and then plugged into the Color Computer Keyboard Jack, and the Keyboard plugged into the other end of the cable, turn on the computer and see if you can "type" to the computer. If there is no keyboard response, unplug the keyboard from the cable and turn the cable over and plug the keyboard back in. If there is still no response, use a continuity checker and trace each of the 16 conductors from the keyboard jack on the mother board to the matching pin at the keyboard end of the cable.

Final Touch:

You may choose to use the CoCo Case to mount your extended keyboard, but then where will you put the computer? OR you may choose to build an enclosure for your keyboard. Even more simple is a pair of wood blocks mounted to each end of the keyboard. They provide functional support at the proper angle and a sharp appearance.



The dimensions are 5 inches long, 3/4 inch wide, 1-1/2 inch high at one end and 3/4 inch high at the other end. Cut a 1/2 inch slot, 1/16 inch wide and 1/2 inch from the top angle cut. The blocks will press fit on the end of the keyboard and can be finished with varnish, varathane or paint. With the extra wires, you can add an LED Power On light and even a reset switch.

-- Rodger Alexander --

Getting Started with MV

It was fall of 1987 when I bought Multi-Vue. I had been using OS-9 Level II on my 512K Color Computer 3 for six months and couldn't wait for Radio Shack to start selling the new desktop interface for my favorite OS. The guy at the store told me that I'd be able to write my own DeskMate with Multi-Vue. I wanted to learn to use icons and click on things rather than just typing commands on boring text screens. I couldn't wait to get started!

That night I read through the Multi-Vue manual and was very excited about creating my own boot disk. I used a backup copy of my Multi-Vue disk to do the grunt work. I executed the script file that would begin the long process of copying most of my system modules in memory to my new modules directory that OS9GEN was going to use. After swapping disks and waiting, I was soon able to boot up Multi-Vue on my system for the first time. I played around with the software until I had done most of what was mentioned in the manual. When I managed to figure out how to start a new shell in another window and I had typed MDIR, I soon realized that I didn't quite have all my boot modules that I was used to seeing in memory. I even had modules loaded in memory that I would never want to use.

After doing some detective work, I discovered that I could create a file in my original modules directory called *multivue.boot* (similar to bootlist created by CONFIG) that could be read by OS9GEN to create a better Multi-Vue boot disk.

Here is how my *multivue.boot* file looked:

```
OS9P2
IOMan
RBF.mn
Ram.dr
R0.dd
CC3Disk.dr
DD.dd
D0.dd
D1.dd
SCF.mn
CC3IO.dr
windint.io
vdgint.io
TERM_win.dt
W.dw
W1.dw
W2.dw
W3.dw
W4.dw
W5.dw
W6.dw
W7.dw
aciapak.dr
t2.dd
printer.dr
p.dd
PipeMan.mn
```

```
Piper.dr
PIPE.dd
clock.60hz
init
```

You'll notice that the main difference between a standard OS-9 bootlist and a Multi-Vue bootlist is that the module *GRFINT.io* has been replaced with *WINDINT.io*. The newer module is a superset of the older module and is required in order to run Multi-Vue. Another thing to note is that there is no *CC3GO* module in the *multivue.boot* file. It turns out that *CC3GO* is really a program, much like *SETIME*, that doesn't belong in your OS9BOOT file. *CC3GO* can be put in the root directory of your boot disk instead.

To create a Multi-Vue boot disk, first insert a formatted blank disk in drive /D1 and your modules disk in drive /D0 (this is the disk that you've always been using to create various boot disks). Type **CHD /D0/MODULES** and **CHX /D0/CMDS**. Then type **OS9GEN 48k /D1 <multivue.boot**. Your disk in drive /D1 will contain the correct OS9BOOT file for Multi-Vue use when *OS9GEN* is finished.

But before you can boot with that new disk, you must create a **CMDS** and **SYS** directory and copy some files and modules to them. You'll need to copy these modules to the **CMDS** directory:

```
shell
gshell
control
grfdrv
```

(These modules are the bare minimum.)

You'll need to copy the file called *env.file* to the **SYS** directory. The files called *stdfonts* and *stdptrs* should also be copied to the **SYS** directory from your master OS-9 system disk. A startup file is also required on your new Multi-Vue boot disk that contains:

```
link shell
sctime </1
date t
merge /dd/sys/stdfonts
merge /dd/sys/stdptrs
control -e
tmode .l echo
gshell<>>>/w&
```

Now you can boot with the new disk. You will find yourself in a */term* window while Multi-Vue is running in another window (press CLEAR to get there). You'll need to add more files and modules to your disk in order to use more of Multi-Vue's features (such as *gclock* and *gcal* and any icon files, etc). But for now, what I have shown you can get you started with Multi-Vue in no time.

I hope that I have simplified the process for creating Multi-Vue boot disks for those of you that have been confused on the matter? I'd like to see more people out there using Multi-Vue. I will say that there are a

few major bugs that can cause system crashes, but there are updates for *GSHELL* available on most BBS's that not only fix the bugs, but also improve Multi-Vue as a whole.

Shawn Driscoll -- Feb. 13th, 1992
Delphi;OS-9 Sig

NOTE: All Multivue patches and update files are also available from *OS9 Newsletter* for the cost of the disk and postage. Send \$1 with your name and address to 3404 Illinois Lane, Bellingham, WA 98226.

Tips from Larry

If you've been putting off getting that 512K ram upgrade for your CoCo, you'll be surprised to hear that CoCo Pro and other third party vendors are out of stock with a long waiting list. But don't despair, I called Tandy National Parts, and they have approximately 800 unpopulated 512K memory boards in stock for \$39.95 each. Stock number is AX 260-3336 for the Color Computer -3, Cat. #26-3334. You'll have to buy and install your own 41256 ram chips on the board, but at less than \$1.50 each, you'll come out ahead!

ALSO

Here's how to make your old original Disk Extended Basic version 1.0 Disk Controller (requiring 12 volts) work with your CoCo -2 or 3. Simply replace the 40 pin controller

chip with an MB 8877 controller chip. Now your old controller will work on the 5 volt only system. 12 volts is no longer required.

ALSO

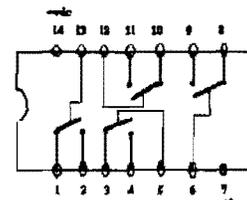
As always, I've got a large stock of *KEL-AM* male and female ribbon cable connects. These are the only connects that can be used to make "Y-Cable" and Keyboard Extension Cables. Let me know your specific needs and I'll make up the cables for you. Call me at (206) 692-5374

ALSO

T. Warren's article in the September '91 Newsletter about installing a Hi-Res Interface Switch was great. But, I was concerned about the relay since most DPDT (Double-Pole/Double-Throw) relays that I

found required a pretty hefty current drain on either the Op Amps and/or the CoCo Power Supply. I called technical assistance available through *Radio-Electronic Magazine* and they suggested that I use a CMOS switch. Either the 4066 or the 4016. Operation is similar to a relay with negligible current drain. to the power supply or bias circuits.

Quad Digital or Analog Bilateral Switch



4066

-- Terry Laraway --

OS-9 Concepts by Kevin Darling

First, shift your thinking to hex and binary. If you're still in decimal, no mercy! Also note this is for the CoCo-3 and OS-9 Level 2 or Level 1 version 2.0

Important signals CoCo --> MPI:

SCS- secondary chip select signal (FF40-5F)
CTS- cartridge enable (8000/C000-FEFF: 32K/16K ROM pak)
Not Used under OS9.

Important signal MPI --> CoCo:

CART- cartridge interrupt, connected to FIRQ via PIA, or
FIRQ/IRQ via **GIME**.
Enabled from slot which also has CTS selected.

As Peter Lyall said, the MultiPak Interface bus is basically an extension of the CoCo bus. Each slot has the same address, data and control lines as another, EXCEPT for the SCS and CTS/CART lines. You may select which slot gets those from the CoCo either by using the front mpi switch during/just after a reset (which switches both the SCS and CART lines); OR by storing a control byte at the FF7F MPI select latch. The control byte looks like:

CTS/CART SCS

CCCC SSSS where CCCC=slot for CTS/CART SSSS=slot for SCS
and CCCC/SSSS are:

(Continued next page)

BIN	HEX	
0000	0	-slot one
0001	1	-slot two
0010	2	-slot three
0011	3	-slot four

(NOTE: They have bits enough for a 16 slot MPI if they wished)

In hex, this works out perfectly. Examples:

For the CART from slot one, and the SCS to slot four, you'd pick 0 and 3 (store a \$03 at FF7F).

For the CART from slot three, and SCS to slot four, you'd use 2 and 3 (store \$23 at FF7F).

WHICH DEVICES NEED WHAT?

Get it? Okay, so how do you know what devices need to be in a certain slot... well, it depends on how well they are internally decoded (that is, do they require the SCS?), and whether they need to send a CART interrupt back to the CoCo:

Some devices on the bus decode themselves fully (such as the RS232 pak, which KNOWS it's at FF68-6B). What that means is, they have all the extra circuitry needed to notice when you're accessing their address. Devices that have this circuitry also include the X-Pad, SSC pak, Modem pak, Wordpak, etc.

Other devices use the SCS-line to help decide for them (such as the Floppy Disk or RS Hard Disk Controllers). Therefore, they will NOT be enabled, unless the MPI select latch is set up to allow the SCS to their slot.

The RS232 or Modem paks need to get an interrupt line back to the 6809 inside the CoCo. Therefore, it's slot must also be selected for the CTS/CART lines before it can get thru. Since only one slot can do so at a time, you have two different drivers, ACIAPAK and MODPAK. ACIAPAK sets up for an interrupt from slot one, and MODPAK instead uses the VIRQ timer to check regularly (called polling) for an interrupt register flag in a fully-decoded device in slots two, three, or four.

So, if a device fully decodes itself (like the SSC pak), it can go in any slot. If it uses the SCS to enable it (like the disk controller), OR needs to use the CART (like the RS232 pak, which is fully decoded), it must go in the slot that the SOFTWARE decides.

Note also, that you cannot use more than one device that fully decodes itself at the same addresses, since such a device is slot-independent in that respect and ALL of them will turn on at the same time: a mess. So you can't use two RS232 paks, for instance, unless YOU internally re-wire one.

Why aren't all devices fully decoded? Cost is why. It's cheaper to make a disk controller that uses the already decoded SCS line from the CoCo, than to include the one or two extra chips needed to decide it's own address. Theoretically, it also meant you could have a disk controller in each slot, and pick between them (which I've done before), but reality strikes.

Why not let all CART interrupts go thru. so that you could use more than one interrupt-driven device? Because the MPI was originally designed for selecting game ROM paks, which used the CTS line to enable the ROMs. after the CART FIRQ had let the CoCo know the pak was there. Many people now tie all the CART lines together for use under OS9, tho; sometimes thru a small circuit, sometimes straight to the 6809.

CHANGING THE SOFTWARE:

In L-I vr2.0, you are expected to power up using the slot in which you have your disk controller. CCDisk does NOT set the MPI SCS latch. However, the ACIAPAK it's associated driver T2, assumes the RS232 pak is in slot one and the disk controller is in slot 4, as it stores a \$03 at FF7F. When the Hard Disk descriptor (CCHDisk, BBHDisk, etc.) needs to access the Hard Disk interface (which uses the SCS line), changes the MPI select to \$02 (CART from slot one, SCS to slot three). So the Hard Drive Interface must be in slot three. In both cases, the RS232 or Modem pak that you wish to be interrupt-driven by ACIAPAK must be in slot one.

The Hard Disk descriptor clears D.DMAFlg when it's not using slot three, so CC3Disk "knows" the SCS is going to slot four when it wants to shut off the motors after a timeout. That is, the disk controller had BETTER be in slot four! If the Hard Disk descriptor is using the MultiPak latch, it sets D.DMAFlg so CC3Disk will reset it's timeout counter and try again later. Drivers that change slots should use this flag byte. Some third-party drivers, most notably the ones by Brian Lantz for Disto and J&M, don't use this flag. In the case of the Disto paks, since they are slot-dependent, these drivers shut off interrupts while accessing the hard disk or ramdisk, while the SCS slot is changed. This is probably in the mistaken belief that drivers can be timesliced (a common misconception) and some other might change slots. There's no reason to do this, unless there is an IRQ-driven device that will change slots on you. So far, I know of none that do.

Personally, I've hacked my drivers to change only the SCS OR CART enables as required and leave the other alone. Since this requires re-assembling the drivers, I can't help you out there. But you CAN patch the Hard Disk descriptor and the ACIAPAK to use a specific slot for the RS232 pak.

The bytes to change are at offsets \$80 in ACIAPAK, and offsets \$4D7 and \$4E9 in Tandy's CCHDisk (if you use it). Currently, these bytes are set to \$03, \$02, \$03. Change the zero (\$0x) to whichever slot code you desire for the RS232 pak, reverify the modules, and you're done.

So, for the RS232 pak (using ACIAPAK driver) in slot three, change them to \$23, \$22, \$23, for example.

USING TWO RS232 PAKS:

The RS232 pak is semi-unique in that it's easy to modify (by design) to a different address. It comes decoded for FF68-6B. But you can change it fairly easily to be from FF6C-6F. Then you can use /T2 and ACIAPAK for the original in slot one. And /M2 and MODPAK for the hacked one in slot two or three. One way to re-wire the

pak is this: Cut the trace running from pin 8 of the 74LS04 (U6) to pin 2 of the 6551 (U1). Then solder a wire from pin 9 of U6 to pin 2 of U1. That's it. That changed the use of the A2 address line. Before it had to be low, now it has to be high. Good luck! Hope this helped out, Chuck. Darn, and only 1400 words, too!

LATE LEVEL-II NOTES:

Ver 2.00.01 of L-II has a bug in it's VIRQ code. To use MODPAK, first change offset \$0C in the INIT module in memory from \$0F to \$0C. Then make a new bootdisk, if wished. If you have problems with your RS232 pak, you may need to install a faster 6551 chip.

Please direct questions to KDARLING - delphi
Kevin Darling 73117,1375 - compuserve os9 forum



Club Activity Report

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

Bellingham OS-9 Users Group

The February meeting was cancelled due to the Bellingham School District cancelling our rental at the last minute. March meeting will be on the 11th at 7:30 p.m. in the Fairhaven Middle School Library, 110 Park Ridge Rd.

--Rodger Alexander--

Longview/Kelso

At our March meeting, we stuffed my CoCo in a IBM/XT case. ran all the ports (serial, parallel, video, and Keyboard. [still working on the joystick]) to the back of the case. Boy, it sure is nice to have all this room on my desk and free plug-in's to use as well as being able to have the keyboard on my lap. Then Allen, removed my ROM chip from the RS-232 pak so we could use it on a Y-cable. then we modified the Disto 4in1 board so it would work right with RiBBS with the auto answer amd carrier detect lines.

We also demonstrated some software and I showed off my new 2MHZ coco to everyone (Them .GIF flicker pictures look nice now. can't hardly notice any flicker at all. The IMG's look real good too, but have a little flicker in them still but look much better).

--Mark Johnson--

Port O CoCo

February meeting of the Port O' CoCo was the best yet. People kept looking around the room with a look of, "I don't know you, do I?" and of wonderment because more and more people kept showing up well after the meeting began. In fact, Port O' CoCo attendance hit a new all time high with 25 people.

With our due structure of only pay as you go we also had a record collection of contributions as well. The publicity is paying off. Phone reminders ably handled by Phyllis Armstrong has made the difference between a so-so turnout and a great turnout. Our phone list is now about forty people from as far away as Olympia and Federal Way. For those people to attend regularly is a show of commitment to the CoCo!

The first and only piece of major business was to decide about getting a CoCo system for the club. A handful of the regulars have been hauling in their system for use and show 'n' tell. This is not a simple task as anyone of us know. By having a simple system we can at least backup disks and show demos. Also it will be easier to bring in a monitor than it will be to lug in all the bits and pieces. There was no opposition to using club money to purchase a system with two drives for this purpose.

Gene Elliott gave a brief report that he has completely transferred the cassette programs we purchased to disk for our club library. Thanks, Gene, for a great job! There were about 100 titles in the collection of games, educational stuff, and various applications.

Our highlight at the February meeting was a presentation given by Steve Fields, manager of the Radio Shack in Bremerton (there is only one there now). Steve's theme was how to get what you want in information, repairs, software, and hardware from your Radio Shack store. "On the wall of each Radio Shack store is the store's creed. At the top of the list is CUSTOMER SATISFACTION. I know I can't give you the world, but I can give you my best."

There are too many employees of Tandy who don't even know that the CoCo used to be part of the Tandy Computer line. And one of the best selling computers. It also is one of the longest living computers with ten years of production. That has come to an end. The CoCo is no longer part of the Tandy line, but that doesn't mean that Tandy has stopped supporting it. Tandy has a long policy of supporting any product for ten years after being dropped from our line. For the CoCo-3 that means that we will repair and offer software for the next eight years.

The Tandy Computer Software Catalog contains almost all of the software ever offered for the CoCo. The drawbacks are that they are being offered at list price, not the \$2-5 prices offered at stores to move the inventory out as fast as possible.

"Repairs are another example of support. We will repair or replace anything we make. And if you have one of our extended warranty programs we will do it at no cost to you."

With these policies already in place then the weak link in the Radio Shack relationship with you is the people you meet when you walk in the store door. They may not know that we ever offered the CoCo. They may not know that we offer a list of scores of software through our national system. They may not know that we stand by our CUSTOMER SATISFACTION POLICY."

Steve wound up his very positive and supportive presentation by stating that if you don't get the satisfaction that you believe you deserve, don't upset the system by calling up the chain of command. That will only make people mad. It would be better to call him and have him work with you through the system. This would allow you to gain two things:

1. A better understanding of corporate Tandy.
2. A workable position of satisfaction for all concerned, including you.

Steve can be reached at (206)373-5029 or 373-1400.

The remainder of the time was in small group discussions and Q&A with Steve. It was an informative and helpful discussion.

In the days that followed we were contacted by a small CoCo group in Yakima. Mike Sanderson (509)453-8598, at 2802 Beaudrey #860-A, Yakima WA 98901, informed us that there is a group of about 5 people who meet to discuss the CoCo. They are looking for ideas and support. Since he has a relative in the Bremerton area he will be attending one of our

meetings in the months to come. In the meantime, if you are so inclined, give him a call and chat about CoCo's. You will discover that he is quite accomplished in Desktop Publishing with a program called C3 Pages.

March's meeting is on the 16th. The topic will be communications between computers and with BBSs. We plan to have a live demonstration of how to sign onto a local BBS.

--Donald Zimmerman--

Seattle BBSxMUG

The February meeting might have been considered an **anti-CoCo** session. Scott Honaker's CoCo was being repaired during February so he was not able to finish up the Basic09 Database project originally scheduled for this meeting. Instead Scott prepared a presentation on the latest hi-tech software/hardware available to the general public and what is in store in the immediate future.

Scott began his presentation by playing some sound files on a 80386 PC equipped with a *sound blaster* card. The "sound" files ranged from 30 to 300 Kilobytes in length and lasted from 3 to 10 minutes. The compactness of the files is the result of the *sound blaster* card reading a single pitch sine wave example for each sound source used then simple performance instructions to the *sound blaster* card. The card then performs the instructions with the sound wave samples.

Scott demonstrated the newest *Windows 3.1* applications, including *Word for Windows 2.0*. This neatly led into a list of projected Microsoft releases for 1992 and '93:

Windows 3.1:

- True Type Fonts
- Faster than *Windows 3.0*
- More robust (fail-safe)
- Multimedia Support

Windows 32 API:

Windows designed for 32 bit bus file server host. Will look and

operate the same as standard windows on the each station.

Windows NT:

Also designed for 32 bit host machine. replaces DOS as the operating system with *Windows 32* and DOS or other operating environment on top. Also referred to as *NT Executive*. It is only 8K in length and can be easily modified/compiled to operate with any CPU including the new high speed MIPS series.

DOS 5.0:

Future releases will include *stacker* archiving utility.

DOS 6.0:

Longer file names (no longer restricted to 8 character file names)

Better memory management to take advantage of upper memory and block switching.

More Utilities (*Stacker*)

Multitasking as the DOS level

Windows centric features (will handle windows better than previous version.)

Disk Compression/Defragmentor included with DOS.

Other juicy tidbits Scott presented are Microsoft's expected standard hardware for the 90's will be 80386 CPU computers running VGA graphics, while the new MIPS type CPU's will be the processors of the future. With the split between IBM and Microsoft and the new partnership developing between IBM and APPLE, the Macintosh computer is rumored to be discontinued within the next couple of years. Microsoft is gearing up to take over support for the Macintosh with the development of OS3. Microsoft will also have the *NT Executive* DOS Shell ready for the Macintosh so they too will be able to run *Microsoft Windows* in their machines and join the rest of the PC world.

Next month Pat Pleuard will present his own Hard Drive Interface.

--Rodger Alexander --

Washington State BBS List

FAR POINT BBS (Seattle)
RiBBS (Fido NET)
(206) 285-8335

COLUMBIA HTS. BB (Longview/Kelso)
RiBBS (Fido NET)
(206) 425-5804

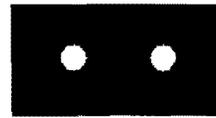
DATA WAREHOUSE BBS (Spokane)
RiBBS (Fido NET)
(509) 325-6787

BARBEQUED RIBBS (Bellingham)
PC-Board (PC-Net)
(206) 676-5787 - CoCo Conference #5

OS-9 TACOMA BBS (Tacoma)
RiBBS (Fido NET)
(206) 566-8857

COCO EXPRESS BBS (Anacortes)
RiBBS (Fido NET)
(206) 293-1057

Color Computer Video Library



Fixing the Multipak "IRQ"
Installing a 2nd floppy drive
Installing 512K Memory Board
Installing a Burke & Burke Hard Drive

\$10

Bellingham OS-9 Users Group
3404 Illinois Lane; Bellingham, Wa. 98226

The *OS-9 Newsletter* is published by the Bellingham OS-9 Users Group. Rodger Alexander, Editor. Publishing software is Microsoft Word for Windows 2.0, printing to a Hewlett Packard Desk Jet 500.

OS-9 Newsletter
3404 Illinois Lane
Bellingham, Wa 98226