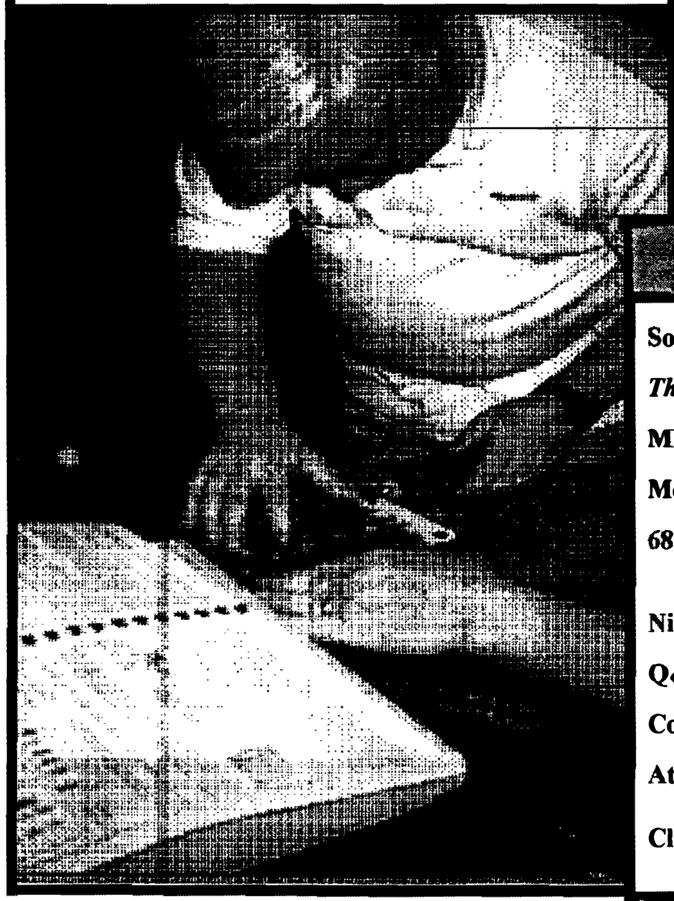
OS-9 Newsletter©

Volume IV Issue 9 Bellingham OS-9 Users Forum September 30, 1993

Mark Kulien installing an internal Modem to Donald Zimmerman's CoCo-3 in PC Tower Case

Solder Fest in Port Orchard

Rodger Alexander cutting multipak to fit into PC Case using a drcmmel roto tool. Note "How To" instructional book.



The Rocket is Dead

Letter of Cancellation from Chris Burke

We have received a number of advance orders for ***The Rocket***, but total orders did not exceed our minimum level of 100 units.

This limited demand, combined with recent price increases from our software vendors and dynamic memory suppliers, have made impractical for burke & burke to produce *The Rocket* at the proposed price of $195. We have considered offereing *The Rocket* with less third-party software, more Burke & Burke software, a higher base price, and no memory, but have decided against doing so since these changes provide a less capable platform and delay the project by several months.

After careful evaulation, we have decided to delay introduction of *The Rocket* indefinitely.

With Our Sincere Appreciation for Your Support,

IN THIS ISSUE

MM/1 Update

This is a general response to the recent inquiries about IMS and it's problems. I am not responding to this as an IMS representative.

First, BlackHawk Enterprises is negotiating an agreement with IMS to assure future production and sales of the MM/1. IMS will continue to sell the MM/1, BlackHawk Enterprises will integrate the systems. This is being done in order to assure two things. First priority is to guarantee production. We want MM/1's available to a rep within no more than 2 weeks. My preference is to have some on site with the rep when possible. Second priority is to keep Paul involved with the MM/1. While Paul has had some problems with management here, he is still excellent at marketing , and we need him doing that for the MM/1 community. With IBM, Apple, etc as competitors, we can't afford to give up anything that can give us an edge, anywhere.

Second, we will discuss the problem of outstanding orders and warranty work. But the amount of debt I am willing to take on to purchase IMS's licenses is limited. I intend to start this venture with our focus on production and marketing, not servicing past debts in whatever form. I think I will have a big enough job making sure I can start out with an adequate inventory, and maintain it, much less sort out IMS's orders. So, I don't intend to ask anyone for proof that IMS owes them anything. I am going to pass these on directly to Paul Ward, and Paul can handle verification. As IMS will continue to sell the MM/1 for us, Paul may choose to handle warranty work on previously purchased machines himself.

Third, we HAVE reached an agreement with a company in Australia to manufacture the MM/1 there. Importing to this country has not been discussed. The margins on the MM/1 are already too narrow to add paying import duties to our expenses. In fact, half of my difficulties in selling my business plan involve coming up with plans to sell packages that increase that profit margin to the point that the banks see us as viable! We will however, realize a certain amount of income that should help us to stabilize domestic production. Further, we assure our Aussy friends a short supply line, and quick delivery, something I hope to accomplish here as well! (Taiwan has not been discussed, because I have no connections there at this time.)

Lastly, I have already begun exploring ways to assure uninterrupted warrantee repair service, assure improved quality of MM/1 production, and decrease lead times. I have been helped in this by Paul Ward, who anticipated several options I had devised. I expect production of 68340 boards to be well underway by the end of the year. This will help to finance MM/1 production, and get us back on our feet.

Presently Paul Ward remains, to my knowledge, the SOLE owner of IMS, and the right to produce the MM/1 under license to Kevin Pease. Your support (either in purchase! or Cheerleading form)is appreciated! Thanks to all who have expressed there support for the sales reps during this time, and please hang in there while we work to get through these trying times!

==David Graham, BlackHawk Enterprises 93/09/06==

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@nwrdc.bones.wednet.edu). Unfor-tunately, 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 admir-ation 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.

More Functions in C

C Tutorial by Randy Kirschenmann

In last month's article we had a look at a very elementary implementation of a C language function call. The *printbits()* function took an integer argument and returned an unspecified and unused integer. We can specify functions that pass any number and type of arguments. The data returned, however, is a bit more restrictive. Only one data element can be passed back to the caller from a function in the C language via the return statement. This may seem to put unsurmountable bounds on our programming efforts using this language. As you will see, however, this is not the case. But, I'm getting ahead of myself again. First we should examine in more detail the fundamentals of C function calls.

When a function is defined to accept a data type as an argument, as in:

```
addint(x)
int x;
{
    x = x + 1;
    return(x);
}
```

the data that the function receives from the caller is a copy of the passed argument. That is to say, in the calling function, such as:

```
main() {
    int x;
    x = 1;
    addint(x);
    printf("x is now %d\n", x);
}
```

you might expect the screen to show:

```
x is now 2
```

I suggest that you compile and test this function. What you will see after running your program will be:

```
x is now 1
```

So what happened? The data that was acted upon in the *addint()* function was not the variable, *int x*, defined in the calling function... it was a copy of *x*. The integer *x*, defined in function *main()* is local in scope to that function. As is the integer *x*, defined in *addint()*, local to *addint()*. No other functions can "see" these local variables. These data items come into existence each time the function is called, and are destroyed when the function is exited. These local variables are referred to as automatic variables, being created and destroyed automatically as functions are entered and exited.

If we change the *addint(x)*; statement in *main()* slightly to:

```
x = addint(x);
```

what is displayed will also change... try this and rerun. You'll get the expected results of:

```
x is now 2
```

In this case the value computed in *addint()* was passed back to the caller, via the *return(x)*; statement, and assigned in the calling function to the variable *int x* in *main()*, which then was passed to *printf()*. In the original example the results of the addition were lost because the caller made no use of the returned value.

Another change which can be made to our example to produce the expected results is to define *int x* as a global (or external) variable. Data defined with global scope can be accessed and altered by all functions within a given program. In order to define a variable as global it is defined outside of any function. Hense, the term external, meaning outside. The variable must also be declared in any function that needs to access it, either explicitly, using the extern keyword, or implicitly, by context. The extern declaration must be used when a global variable appears in a function, within the same source file, before its definition. It is common practice to define the global variables at the start of the source files, thus eliminating the need for the extern keyword in the declarations. When compiling

mmultiple source files, global variables declared in more than one source file must have the extern keyword appended to the declaration.

In our example we would write:

```
int x;              /* notice where this is placed */
main() {
        x = 1;
        addint();
        printf("x is now %d\n", x);
}
extern int x;       /* strictly speaking this line is not
needed  */
                    /* unless addint() is coded in a
separate file */ addint()
{
        x = x + 1;  /* here x is declared by context */
}
```

We see, by running this program, that *x* is acted upon directly within the *addint()* function. However, instead of operating on any int data item which we choose to pass to *addint()*, now we are restricted to the single globally defined data item, *int x*. Global data can be used instead of argument lists to

Continue Next Page

More Function Calls in C *Continued*

ommunicate between functions. Generally, though, the liberal usage of global variables is not recommended. Locally defined data is safer. Only the function in which it is defined has access to it. No outside routines will ever corrupt local data items.

Another method of communicating between functions is the use of pointers. Returning to our original function let's alter the parameter list slightly to accept a pointer to int, rather than an int data item.

```
main() {
    int x;
    int *iptr;          /* define a pointer to int */

    x = 1;
    iptr = &x;          /* set pointer to int variable's address */

    addint(iptr);       /* call function, passing address of x   */
    printf("x is now %d\n", x);
}
addint(x)
int *x;                 /* redefine argument as a pointer to int */
{
    *x = *x + 1;        /* this will add 1 directly to address passed */
                        /*  to our function via the pointer        */
}
```

By running this version we see that now *addint()* has direct access to our int data variable, passed via the pointer. This allows us to directly manipulate data defined as local to a calling function from a subroutine. Notice that the variable *x* in *addint()* is not the variable *x* found in *main()*. In fact the variable x in *addint()* also is not the variable *iptr* in *main()*, which is an address. What is given to *addint()* is a copy of *iptr*. The actual value in *iptr* is not affected by *addint()*. If we were to write a more complicated function involving passed pointers we can demonstrate this. Let's define a function which will parse an array of integers and add all of the members together.

```
addarr(iptr, n)
int *iptr;
int n;            /* this will tell us how many items are in the array */ {
  int i, total;
  for(i = 0, total = 0; i < n; i++, iptr++)
     total += (*iptr)++;
  return(total);
}
```

In this example, notice the format of the *for(;;)* statement. I have used the comma operator to perform a dual initialization of the two variables ***i*** and ***total***, and a dual incrementation of ***i*** and the variable ***iptr***. In C a pair of expressions separated by the comma operator is evaluated left to right, and the type and value of the result are the type and value of the right operand. This allows both *i = 0* and *total = 0*, as well as *i++* and *iptr++* to be "evaluated" in the same expression.

This function could also have been written, using array notation instead of pointer notation, as:

```
addarr(iptr, n)
int iptr[];
int n;            /* this will tell us how many items are in the array */ {
  int i, total;

  for(i = 0, total = 0; i < n; i++)
     total += iptr[i]++;
  return(total);
}
```

Notice the slight differences in the two versions. The array notation may be a bit easier for the novice C programmer to understand, but the pointer notation is considered to be more elegant and is preferred by experienced users of the C programming language.

Also of interest in this example is the expression "total += (*iptr)++;". To understand what is happening here we need to understand first that **iptr++* means to increment the value of the pointer, not that to which it points. Whereas *(*iptr)++* means to increment the value found at the address which is pointed to by *iptr*. This is due to the hierarchy of the operators defined in

Continued Next Page

More Functions in C *continued*

C. Each operator used in C falls into a specified hierarchy; that is to say that the operators with a higher hierarchy will be evaluated before those that fall below it. Most operators of equal hierarchy are evaluated left to right, or except for a few that are defined as being evaluated as right to left. The precedence and order of evaluation are given in the following table:

Returning to our example, since we are using post increment, this increment will take place after the pointer is used in the expression:

```
    total += (*iptr)++;
```

Thus, what we have coded on this line says: add to total, the value found at the address pointed to by *iptr*, then increment that value. Our *for(;;)* statement will loop thru ***n*** times. At the *n*+1'st iteration of the loop our conditional expression, *i* < *n*, will be false, causing control to exit the loop. At this point *iptr* will be pointing to *(length-of-int-type * n)* bytes higher in memory than it was when we entered the loop. Our function will return to caller the total calculated.

To inspect the values stored in our array of *int* we will write another short function:

```
   printarr(iptr, n)
   int *iptr, n;
   {
           int i;
           for(i = 0; i < n; i++)
                printf("arr[%d] = %d\n", i, *iptr);
           return;
   }
```

Now we write *main()* to demonstrate what we are discussing:

```
   main() {
           int *iptr;          /* this pointer is needed only to demonstrate */
                               /* how pointers can be passed to a function   */
         static int arr[4] = {1, 2, 3, 4};
           iptr = arr;         /* this initializes our pointer */
           printarr(arr, 4); /* we can use the array name as a pointer */
           printf("the sigma sum is %d\n", addarr(iptr, 4));
           printarr(arr, 4);
           printf("*iptr = %d\n", *iptr);  /* where is iptr pointing now? */
   }
```

First off you'll notice that I've declared my array of *int* to be static. K&R C will not allow us to initialize a local array in the declaration. I could have specifically assigned values to each of the array elements as in:

```
    arr[0] = 1;
    arr[1] = 2;
```

etc. But by declaring it as static the compiler will tag this as external and thus allow us to initialize it in the declaration. Static local variables, though external in linkage, as are global variables, are still hidden from other functions. Global variables declared as static are visible only to functions within the same source file. This adds an additional level of privacy to our data declarations. A static variable is initialized only once and retains its value even when the function within which it is declared goes out of scope. What this means is: if we call a function which contains a static local variable, after returning from the function if we call that same function a second time the variable will still have the value it had when the function was exited. More clearly, compile and execute this program:

```
   main() {
           int x;
```

Operator	Associativety
() [] -> .	left to right
! ~ ++ -- - (type) * & sizeof	right to left
* / %	left to right
+ -	left to right
<< >>	left to right
< <= > >=	left to right
== !=	left to right
&	left to right
^	left to ight
|	left to right
&&	left to right
||	left to right
?:	right to left
= += -= etc.	right to left

Continued Next Page

More Functions in C *continued*

```
        for(x = 0; x < 20; x++)
                printf("x is now %d\n",f());
}
f() {
  static int i = 0;
  return i++;
}
```

Note that the value being displayed is not the *int x* defined within *main()*. This variable is used only as a counter. The value displayed is the *int* being returned to *main* from *f()*. If we remove the static qualifier, recompile and then execute this program we see a different display. Running the modified code demonstrates the action of the static qualifier for single source file programs.

Our program initializes a pointer variable with the address of an array of *int*. This pointer is now synonimous with the array; that is to say that the pointer variable and the array name point to the same address (remember, the name of an array is a pointer constant). The first exeecutable statement is a call to our print function which will iterate through the array and print each of its values. Following this, the sigma sum is computed by a call to the function *addarr()*, and the result returned is displayed via a call to *printf()*. Notice how all this can be accomplished in one line of code. The expression "addarr(iptr, 4)" evaluates to the value returned by the function when it is exited. Also notice how the pointer passed to this function is incremented within the function to access each of the separate elements of the array. However, upon exiting, the pointer used in *main()* still retains its original address, that of the start of the array. We verify this in the final statement of *main()*. Before this statement another call to *printarr()* shows that each of the elements in *arr* has been incremented by *addarr()*.

What this program shows us is that elements of an array can be altered permanently from within a subfunction. The subfunction has access to the actual addresses of each of the elements and thus has the ability to effect a permanent change upon the data stored at that address. A word of caution is in order here. *Printarr()* and *addarr()* have no idea of where the array starts or stops. We pass a pointer and an integer value to these functions to be used as the lower and upper limits to their iterations, but there's nothing to prevent us (or another user of these functions) to send erroneous values, ones that would cause *addarr()* to increment data at addresses outside the bounds of our array. In fact we can access memory that doesn't even belong to our program's allocated memory. This places the responsibility directly on the programmer to properly use pointers. The C compiler has no safeguards to prevent illegal use of pointers and if our programs produce spectacular crashes, this is generally the first thing to look for.

So far we've seen only a few of the many various methods available to this programming language of defining data and acting on it. We haven't looked at *structs* and *unions* yet, which provide the C programer with the means of defining new and useful data types akin to Basic09's type statement. As we expand our knowledge and command of the C programming language we'll really begin to see how versatile and powerful a tool it can be.

==Randy Kirschenmann==
Mt. Rainier CoCo Club
FioNET;OS-9 Echo

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

CoCo Hardware For Sale

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

2. I have a file in my */DD/SYS* directory called */DD/SYS/Shell.Parameters* . This is a special script that *Shell+* reads and executes when it starts up. Following is it's contents:

```
    r=</dd/user/trix/startup -p
```

What this does is to tell Shell+ to redirect STDIN to come from the script file */DD/USER/TRIX/startup* and to not display a prompt.

3. Next I created the file */DD/USER/TRIX/startup* as my own personal startup file. And here is what's in _that_ file:

```
    path=/dd/cmds /dd/etc
    chx /dd/cmds; chd /dd/bbs.dev/daemon
    p ; i=/1
```

The first line of this file sets up my path (insuring that /DD/CMDS will always be searched for executables no matter where I CX to). The second line sets up my execution and working directories. The third line does two things; first it tells *Shell+* that it's ok to show the prompt again, and second it redirects STDIN "back" to the same path as STDOUT.

==John Farrar==
FidoNET;OS-9 Echo

A: Phil, I use a special version of CC3GO by Roger A. Krupski. What it does is set your default CMDS and working dirs before exiting the startup file. For my other windows I usually change PWD and PXD to the desired dirs and then start a shell ie, **CHD /dd/games;CHX /dd/games/CMDS;shell i=/w[1,2,3,etc] p="OS9[@]_\$:"&** . All handled in startup.

CC3GO - Found in CC3GO.(AR or PAK) from Roger A. Krupski. CRC = 8E7E34

This gives Shell Plus users the option of using a parameter file to define startup paths and gives back 256 bytes of wasted system memory. I like this one myself!

NOTE: DO YOURSELF AND ANYONE ELSE YOU NEED HELP FROM A BIG FAVOR... GET SHELLPLUS version 2.1. I can't begin to tell you what you are missing!!! There is also a two byte patch for the version 2.1 Shell.

==Paul Fitch==
FidoNET; OS-9 Echo

☺✌☺☞ IT WORKS ☜☺✌☺

This is the third time that we have printed this project in the ***OS-9 Newsletter****. The original article was submitted by Pat Plueard and demonstrated at the Seattle 68xxxMUG meeting. We gladly published the article in the May '92 issue. However we could not duplicate Pat's success and determined that the logic circuity for the I/O Address was incorrect. Unfortunately, we incorrectly used the PC AT Hard Drive Address instead of the XT Hard Drive Address. We also published our findings before we had a chance to test our re-design. For this we are very sorry. We know that we have caused failed efforts and frustration to several of our subscribers (we recieved comments). Our own frustration led to putting the project on the shelf for nine months. In May, mostly due to the request of* ***Mark Kulien, we once again tackled the interface project, but this time with the assistance of*** *Mike Pleas* ***and this time we got it right!*** ***IT WORKS!!!!!***

Figure 1 is the circuit diagram for the interface card. It's very simple and can be accomplished using point to point wiring with 30 guage wire such as that used for wire wrapping (R.S. Cat.# 278-501). Three integrated circuits are used: 74LS32 Quad 2-Input OR Gate, and two 74LS04 Hex Inverters. The chips are very inexpensive and readily available from any electronic supply store. The most difficult part of the construction is soldering the 62 position PC/XT Bus Card-Edge Connector (R.S. Cat.# 276-1453) to the end of the circuit board. Refer to the *May '92 OS-9 Newsletter* for the IBM Bus pin-outs. To simplify the construction, use a pre-punched plug-in circuit board available from Radio Shack (Cat.# 266-192). This board has 72 position card edge connectors, we only need 40, so carefully cut off the unnecessary connector traces making sure that the remaining 40 traces will line up properly when plugged into the Multi-Pak.

The only remaining hardware is an XT-MFM or RLL Controller Card which retails for about $50. I've also purchased several cards recently at the AM Computer Swap Meet for $20. Specifically you are looking for a Western Digital **WD1002** series or the **WDXT-GEN** Controller Card. You can also use the **DTC 5150CRH** and **5160CRH** or Adaptec's **2072.** Just make sure that if you are going to use an RLL Hard Drive that you use an RLL type controller. Some MFM Hard Drives can be made to work with an RLL Controller, but you're pushing it! (See Burke & Burke's CoCo XT Manual, pages A16-A20 for a more detailed explanation)

SOFTWARE:

This project is almost too good to be true. For less than $10 in parts you can build your own hard drive interface that looks very similar to the Burke and Burke Interface, and in fact it operates under the Burke and Burke Software. However, be aware that B&B's software package is copyrighted and you will need to purchase the software from Burke and Burke (P.O. Box 733 Maple Valley, WA 98038. 1-800-237-2409 or 206-432-1814). Burke & Burke will sell the software separately for about $10.

Continued Next Page

Puppa PC Keyboard Interface to be published in *68' Micros*

68' micros has been a success so far! The second issue will be printed and sent by the end of this week. FARNA Systems will also be at the CoCo Fest in Atlanta. The SEPT 15 issue has the schematic for the Puppo keyboard interface. The NOV 1 issue will have a source for bare boards for those who wish to build their own. There will be support coming for G-Windows and industrial OS-9 in general. Support is also there for DECB, OS-9 (CoCo), and OS-9/68000. We are delivering a 30 page 8.5"x 11" magazine.... it is literally STUFFED with information! 68' micros is published 8 times per year (every six weeks, approx.). Yearly subscription is $23, six months is $12. Send subscriptions to: FARNA Systems, Box 321, WR, GA 31099-0321.

I will be sending the 15 SEPT issue out for all new subscriptions until 01 October. Copies will be available at the Fest, as well as copies of the first issue (01 August). Back issues are normally $4.95 for the first, $4.45 each additional. For those subscribing before 1 October, the first issue will be available for $3.50 (the price they will be selling for at the Atlanta Fest), including shipping.

We have about 175 subscribers now (all PAID subscriptions!), and expect to have well over 200 (close to 250) by the end of the year. For those who have subscribed already, thanks for the support!!

==Frank Swygert==
FARNA SYSTEMS

NitrOS-9 Bench Test Update

A friend of mine just called me telling me about the newest issue of the *No Name Magazine* he had just received. I have not seen it yet, but there was an article of some interest to me. It was an article about *NitrOS9* and *Power Booster*, comparing speed increases over stock *OS-9*. The article was written by Alan Dekok.

I'd li

mentioned in the article. Apparently the article did not mention the *NitrOS9* version used. As far as I know, it was v1.07, the version purchased by Alan. The times for *NitrOS9 v1.15* are a little different, as you will see...

These tests were the time taken to list a text file. I cannot remember how big it was, but the same file was used in each test.

Screen Type	Stock OS-9	Power Booster	NitrOS9 v1.07	NitrOS9 v1.15
80 Col Text	41	36	28	20
80 Col Graphics	96	75	52	37
4 Color Grf.	258	205	169	127

These times are in seconds.

==Wes Gale==
FidoNET;OS-9 Echo

Q&A

Q: How can I force OS-9 to boot up to another directory besides /D0 and /D0/CMDS?

==Phil Lewis==

A: You could stick *login* as the last command in your startup file. That way when you logged on to your system, it

from the /dd/sys/password file. Check the manual on login before you do this, so that you don't accidently lock yourself out of your computer!

==Colin McKay==
FioNET; OS-9 Echo

A: Another solution is to create a new window and start a shell, they will have the directories of the window where they were created. (the Parent and Child stuff). So if you want two windows with the same directories, create the new window while in the window with the desired directories.

I think that you will also find that in your startup file you can;

```
echo Initialize Window 1
load wcreate (not required if wcreate in memory or in
/d1/cmds)
chx /d1/cmds
chd /d1
iniz w1
wcreate /w1 -s=2 0 0 80 25 5 1 1
shell i=/w1&
```

W1 should have D1 as its data and exec dirs. This will not affect the Term window, since when the startup file ends, the shell that it is executing from, will die (in OS9 speak). The 'term' window directories are set in CC3go.

==Merv Curly==
FidoNET; OS-9 Echo

A: You can have OS-9 start up in a directory other than the root of your boot drive. Here's how I did it:

1. Get rid of the stock shell. :-) Get your hands on a copy of *Shell21.AR* (*Shell+*).

Continued Next Page

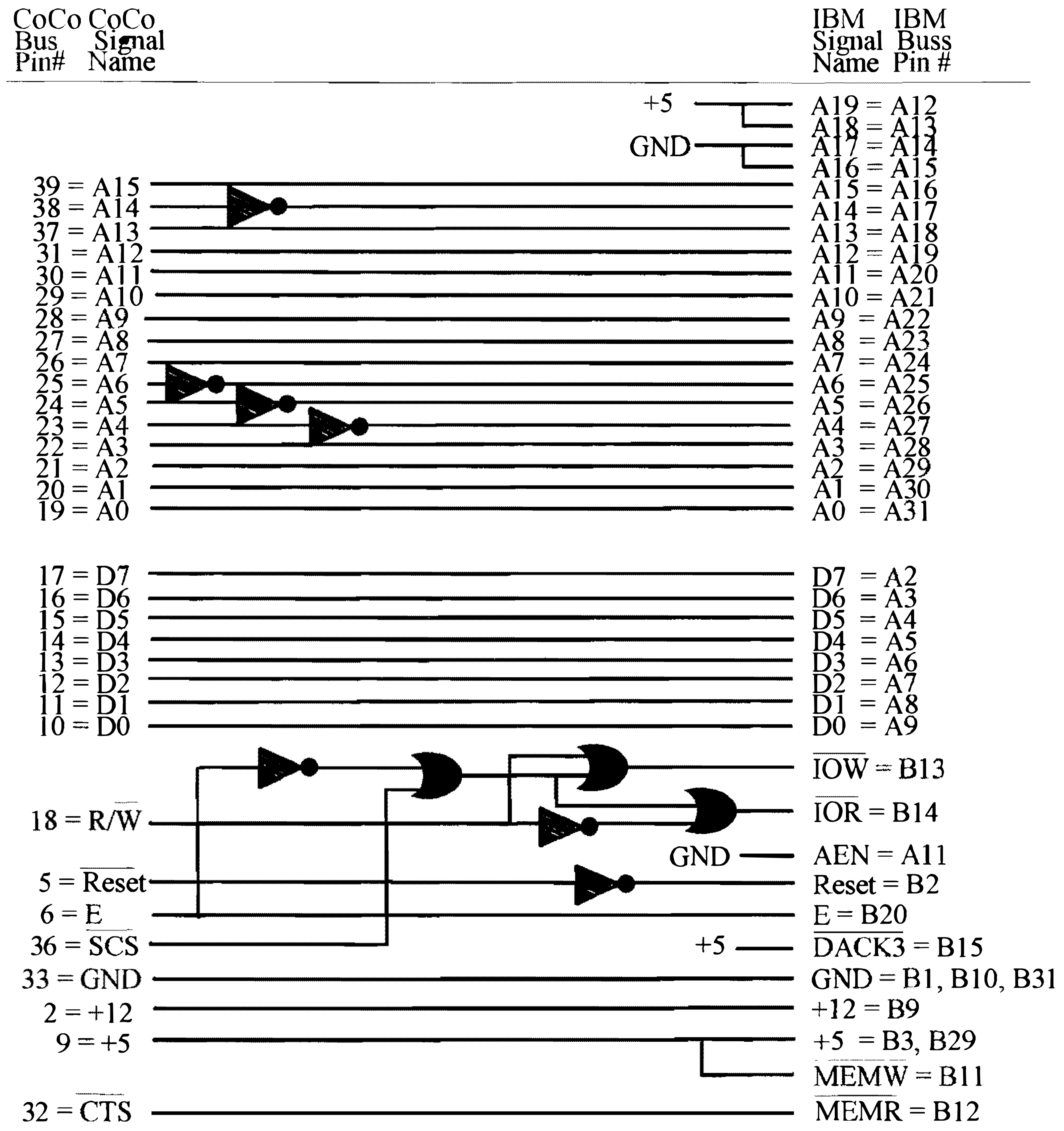


Figure 1

Would You Believe It?

FREE ADVERTISING SPACE
TO SUBSCRIBERS

Club Activities Report

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

Seattle 68xxx MUG

The September 7th meeting of the Seattle 68xxx Micro Users Group began with a demonstration of the successful completion of the parallel port first featured in the February '93 issue of the *OS-9 Newsletter*. After about 10 minutes of creative patching and stretching of power cords, we finally had a CoCo-3 running on a monochrome monitor with a disk drive and a Tandy DMP-105 Printer hooked up to the serial port on the CoCo set at 2400 baud. Another DMP-105 Printer hooked up to the new DB-25 pin **"Parallel Port"** located on the back of the CoCo just below the Joystick Ports.

Rodger Alexander removed the cover so we could see how a second 68B21 PIA chip was "piggybacked" on top of the original PIA on the CoCo's motherboard. The wiring for the parallel port all came off of the left side of the chip and was routed under the mother board to the back of the CoCo where a DB-25 pin female connector was mounted on the lower portion of the case.

We had a "Parallel" vs. "Serial" race. Both printers operated at the same time although there was a marked difference in performance between the serial operation and the parallel operation.... You guessed it, the parallel printer finished way ahead of the serial printer even though the serial printer started first. When the parallel printer ran out of paper, an error message was displayed: **"Error 199 - Paper Out"**.

Scott Honaker diagrammed on the chalk board the difference between parallel and serial information sent or received and how the original PIA was configured by Tandy to work as a serial port so that the two sets of data lines could be used to pass information to two joystick ports, the cassette port and the "bit-banger" serial port. The idea was to get as many port functions as possible from the least amount of electronics.

Donald Zimmerman gave a report on the "Solder Fest" held at Port Orchard on August 19th (See the August issue of the *OS-9 Newsletter*). Photographs of the event were passed around as well as pictures of last month's meeting and The Mt. Rainier CoCo Club meeting and the Port O'CoCo club meeting.

Donald also reported that the Port O'CoCo Club was underwriting the cost of setting up an informational booth at the next two A&M Computer Swap Meets (September 11 and October 2nd). Volunteers are needed to "man the booth" for both dates. The Computer Bank Charity will also be present at both Swap Meets. They have lost storage space at the Fir Crest facilities in Seattle and will be trying to "unload" as much PC equipment as possible. *They won't be undersold! Bargains to be had!* Etc., etc.

Donald shared the latest issues of "*The International Underground*" featuring an article about "The Rocket" by Burke & Burke. Discussion followed as how to notify Chris Burke as to your support or lack of support for the increased cost ($50) for "The Rocket". Chris appears to respond almost instantly to his internet E-Mail address: **burke@mdd.comm.mot.com.**

Bob Belka shared his "free" purchase of a keyboard from **Vetco**. It was especially notable because the case fits the CoCo Keyboard exactly. **Vetco** is an electronics surplus outlet located on Northrup Was in Bellevue.

Scott Honaker shared recent information regarding Motorola's development of the **"Power PC"** RISC processor family based on parameters supplied by IBM. The result will be a new generation of Macintosh type computers that Microsoft will also provide "Windows NT". As a result, the Power PC will have access to all Macintosh software as well as all Windows applications.

The final portion of the meeting was provided by **Buzz Jones** who brought the "Dysun Alignment Disk" along with the J&M Utility package that accesses the alignment disk. We noticed some problems with one of the disk drive we were using so we ran a "Quick Test" and discovered that the second head of Drive 1 was badly out of alignment. This is an amazing package that is absolutely necessary for repairing disk drives. Unfortunately, the Dysun Alignment Disk cost about $100 for the 360K format.

==Rodger and Barbara Alexander==

Port O-CoCo Club

This month the Birthday list consisted of **Chris Johnson** of OS-9 Tacoma BBS fame, **Richard Ambler** and **Bud Helch**. We hope they each had a healthy and happy day! We had no downside this month. We are unaware of any of the group being under the weather.

First on the agenda was the topic of the A&M Computer Swap Meet. We had purchased table space from them to promote the club. We were next to the Computer Bank Charity and our corner of the world brought in a lot of interest.Terry was the hub of activity with his supped up CoCo and a hot color printer cranking out zinging graphics of all kinds and colors. Even those high tech coloraser and bubble ink jet types are impressed by what the CGP-220 produces under the tight command of Terry!

Right next to our booth was a dealer selling mini towers cases w/power supplies for only $35 (*TLC Communicatis, 1045 Wildwood Blvd SW, Issaquah WA 98027-4506, Phone (206) 392-9592*). Accorng to **Rodger Alexander** the placement of the slots, the power supply and the width of the tower is just right for our conversion to a tower setup. At the Seattle meeting this month another source of towers was mentioned: *PC Workers at 13219orthup Way #110 in Bellevue. The phone is 641-4141.* Their mini towers were $40 without LED display with power supply. Prices sure are coming down from the $100 plus just a year ago!

Port O'CoCo will offer a series of three presentations during our October, Novemeber and December meetings. There will be two topics. The first will be a 40-50 minute presentation about BASIC and RS-DOS. Presented by **Gene Elliott**, the presumption will be that you have no previous experience with either. This "*Starting from the Beginning*" presentation will be during the first part of the meeting. After a brief break the second presentation will be on OS-9 and the applications available under OS-9 **Mark Kulien** will be one of the presenters. The presentation will also be about 40-50 minutes long. That will bring us up to about 9 p.m. and the rest of the time is for informal help and discussion. Everyone is welcome regardless of your level of understanding.

==Donald Zimmerman==

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 specifiy OS-9 or OS9/68000!

Bob van der Poel Software

PO Box 355	PO Box 57
Porthill, ID	Wynndel, BC
US 83853	Canada VOB 2NO

Phone (604)-866-5772

Atlanta CoCoFEST

I thought I would briefly take the time to list some of the new goodies I saw at the fest in Atlanta the past two days...(October 2 & 3)

Graphical adventure game of last year's Atlanta fest... digitized pictures of the last fest make this adventure game great! Available for OS-9/CoCo and OSk/MM1.

TOWEL: A great menu-driven disk utility for easy copying, deleting, moving, making, and zillion other things with disk files... very slick. Available for OS-9/CoCo and being ported to OSk/MM1.

GCal: a Graphical Calendar program by Bill... sorry I forgot his last name! He always works with IMS or BlackHawk enterprises for MM/1 sales. The program, then, is for the MM/1.

Calculator: A free desktop calculator was given away for MM/1 K-Windows. Bible oncordance... a very very fast and slick bible concordance was being sold for OSK... a coco/OS-9 release date was set for two weeks.

RS-DOS term program for OS-9 codes: If you want to access an OS-9 BBS or just access another OS-9 computer from the serial port, but still get OS-9 codes (colors, overlay windows, underlining, reverse video, pulldown menus, etc) from RS-DOS, there was a great program for RS-DOS that even ran on a tape (non-disk) system to emulate all those nice OS-9 codes!

MM/1 340 UPGRADE: Makes the MM/1 really fly! This was an amazing thing to see... the graphics on the MM/1 were very very fast! Almost a must for MM/1 owners.

SpeedDisk - OSK: (and maybe CoCo OS-9??) disk defragmenter. Really speeds up disk access for those with hard disks that are experiencing fragmentation!

GNOP: (PONG spelled backwards!) Chris Hawks of Hawk Soft wrote this neat little program using MM/1 hardware scrolling (!!) for a neat pong game where the ball stays stationary while the screen moves around the ball, and you have to line up the paddle to hit the ball!

KVed: A wonderful and dazzling front-end for MM/1 version of VED! Buttons and stuff make editing text files a delightful process!

Alan Dekok's Smash Game: Drew much interest, as it ran with full arcade quality under CoCo OS-9! Amazing speed and graphics!

Also: **A new OSTerm for MM/1; fax display program running under GWindows.; fax possibly on the CoCo; Truetype fonts on the MM/1; MM/1 screen button designer; termcap version of OSTerm; termcap version of Write-Right; MM/2! 68030 upgrade board**

Ok, I am at a loss for more rumours, although there are more, so I'll leave the rest for others to try and fill in!

==Mathew Hegberg==
" <JOELHEGBERG@DELPHI.COM>"

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

OS-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

OS-9 Newsletter
3404 Illinois Lane
Bellingham, WA 98226-4238