

CoCoNutz! E-Zine



Volume 3 Issue 1
February 2007



www.Coco3.com

THE ASIMOV AWARDS

Some 15 years after the CoCo3 was supposed to die, it's still alive. There is an active and creative community, with new hardware showing up every few months, and the NitROS-9 D.O.S. being constantly updated and bugfixed. A mailing list, plenty of websites and groups, and even a newsletter, the picture seems quite healthy.

But something is missing...

Seems that most of the software coming out is for developers, and not much else is being developed.

It sure would be nice to have some of the great programmers that have worked with the CoCo do some new magic, but the real life fact is that writing a new game is a lot of work, and even one of the best ever games for the CoCo, "Gate Crasher" just sold some 40 copies.

So here is something to encourage programmers a bit.

The "Asimov Award" will be given once a year to the best CoCo program for the final user.

The prize is only \$100, and I know that nobody is going to quit a day job to go back to the CoCo, but may be YOU are on the edge, just waiting for another small excuse to go back to the keyboard, and this might be it.

RULES:

The rules are simple. Just email me your program - a DSK file would be great - and you are good to go.

The program has to be:

ORIGINAL; meaning that you didn't copy it from a magazine or any other source, even if you modified it. It can be a clone of an existing program, as long as YOU write it, or port it to the CoCo.

UNRELEASED; it was never available to the general public. We want NEW programs.

COCO 3 COMPATIBLE; the only requirements to run it should be a standard CoCo 3. 128 Kb, 6809, and joysticks. It can be a disk only program, and use any of the display modes available.

DEADLINE:

The program must be sent no later than January 31, 2007. The winner will be announced no later than February 28, 2007.

That should be all. I would really like to be able to offer for sale the winning program, and any other entries that will fit in a standard CoCo diskette for \$5 + S&H, and use part of the money to finance a bigger prize for the next award, but that's up to the participants.

Deadline Extended!!! Now you have until Feb. 15

The winner will be asked for an interview in the next issue.

Visit <http://coco.sclaudia.net> for details!!



Daddy's CoCo stuff as listed in CFDM:

- 1) a COCO 1 with 64k and what looks like a cable for hooking up a composite monitor.
- 2) a COCO 2 (64k) & 3 (512k)
- 3) FD-500 & TWO FD-501 controllers + 4 FD-501'S
- 4) FD-500 CASE & TWO FD-501 CASES
- 5) upgraded #26-3124 MULTI-PAK
- 6) DWP-230 & DMP-105 PRINTERS
- 7) DC MODEM PAK (#26-2228)+ ORCHESTRA-90 CC
- 8) 4 JOYSTICKS + MOUSE & MOUSE PAD
- 9) CM-8 RGB COLOR MONITOR + 13" COLOR T.V.
- 10) various games ETC on ROM PAKS, TAPE, +DISKS

My Dad and his Mom (1984?)

```
0 PMODE1,1:PCLS:SCREEN1,1:FORX=0TO65536:FORY=0TO50:R=RND(129)-  
1:S=RND(97)-1:C=RND(4):COLORC:LINE(R,S)-(255-R,191-  
S),PSET,B:NEXTY:PCLS:SOUND160,1:NEXTX
```

Here is a short one originally submitted to The Rainbow by Greg H Taylor of Naperville, IL. Kaleidobox was published in May 1985 issue of The Rainbow as well. I like this one because it illustrates a creative way around a problem. Instead of trying to replicate to four identical quardrents, wich would be slow, the author kind of sidesteps the whole issue. He does it using one of the CoCo's built in commands, achieving a certain economy, and literally thinking outside the box!

(Sent in by The Captn')

INTERVIEW:

Allen Huffman

Allen is a long time CoCo enthusiast who up until recently was employed by Radisys (who purchased Microware). He did support and maintenance for OS-9.

One of the team at Sub-Etha and a keen Hitchhiker's fan, he has found the time in his busy schedule to answer a few questions for this interview.

(Interview by Nickolas Marentes – JAN 2007)



You were very active in the CoCoFest scenes, even compiling a book about them.

- What was the first CoCoFest you ever attended?

CoCoPro's 1990 Atlanta CoCoFest. I didn't have a car, and my partner Terry didn't have anything reliable. And we were broke. Dave Myers of CoCoPRO! put me in touch with a lady in Houston (my original home town) and somehow we arranged to hitch a ride with them. She and her husband drove a few hours North to Lufkin where we were, and then we rode in the back of a covered pickup truck the entire way to Atlanta -- about 12 hours.

- How many have you attended?

I attended all five of the Atlanta CoCoFests, the one Mid America Fest in Iowa, three Pennsylvania Fests, and all but one post-Rainbow Chicago Fest so about twenty two or so. There was a Pacific Northwest CoCoFest that I also missed, but I think I've made just about everything else.

- What were your greatest memories/experiences of attending a CoCoFest?

The friendships are the most important thing, and meeting all the "hey, I know your name people!" The top ones were Steve Bjork and Marty Goodman, two of the most famous CoCo celebrities that I've ever seen at a CoCoFest. Every CoCo kid knew Bjork from seeing his name on all those games, and Marty Goodman was a staple of Rainbow Magazine. Although I think I only met Marty one time, years later I consider Steve one of my close friends.

- Which was your favourite CoCoFest?

I think it was the 1993 Chicago Fest where we had folks from Europe and Japan all join up for a global OS-9 meeting. Microware was even there with a booth, selling copies of OS-9000 for the PC at discount rates. I think that was the most substantial show I attended, though there were some other amazing ones. I remember Boisy Pitre wowing us all by showing how he could boot a consumer CD-i player in to an OS-9 prompt with a special CD-ROM he made. There were plenty of little things like that, but I think the world meet was a big one. You haven't seen OS-9 until you've seen ERROR #216 in Japanese.

- Is your CoCoFest book still available for purchase?

I actually do have some, and I even know where they are, but I haven't had any way for people to get copies. But, yes, if someone wanted one, I could provide it.

- How do the CoCoFests of recent times compare to the fest of old?

They are much smaller, though I bet most of the people I'll see there this year were at shows five or even ten years ago. It's down to the real die-hards, but so many friends have moved on, or in some cases, passed away. That's the biggest change -- NOT seeing someone who's always "been there."

Tell us a bit more about Sub-Etha.

- Where did the name come from?

The Hitchhiker's Guide to the Galaxy by Douglas Adams used "sub-etha" as kind of a sci-tech running gag. There was the "sub-etha sens-o-matic" and dozens of other "sub-etha this" and "sub-etha that" references. The original name for the company was going to be "42 Technology" also from Hitchhikers, and we even had our first ad printed out and ready to be

sent in to Rainbow. I'm not sure why we changed it, but I know Rainbow offered us a special deal to move from this tiny business card sized ad to a 1/4 page (or maybe it was just 1/8th) for the same money so perhaps we just had more room for a fancier ad and chose a fancier name.

- Were you a sole operator or partnership?

It was a partnership between myself and Terry Todd. I forget the exact type of business, but I know it had the word "Limited" in it and Terry always thought we should incorporate or something so we'd sound less "limited."

- How did the business go and is it still operating today?

We couldn't quit our minimum wage day jobs, but it made enough money to pay for us to attend so many CoCoFests. So many times I literally headed off from Lufkin in my Honda Civic with only a Philips 66 gas card and \$20 cash, hoping I'd find stations that accepted that card. We totally relied on making money at the show to make it home. It was so bad, we'd literally check in to the hotel for Friday night then have to go to the hotel front desk on Saturday afternoon to pay them for the first night.

Eventually I got credit cards. How I wish I hadn't.

- What was your best selling product?

MiniBanners, which was a banner printing program that didn't use graphics modes. It could print to any printer - daisy wheel, thermal, etc. Some printers could print blocks so those could be used. For a daisy wheel printer, you could print banners using X's or *'s or whatever. It also printed multiple line banners and used CoCo RS-DOS fonts. The resolution of large letters was not great, but it did a great job for what it was designed to do. We used it for all our booth signs, and I was very happy when I saw Adventure Survivors making use of it the same way. It was ported from RS-DOS to CoCo OS-9, then later the MM/1. The CoCo OS-9 version was the best seller, and it let you print banners while you did other things with the computer. Gotta love multitasking.

Tell us a bit about Microware.

- What was your role there?

I was hired to teach OS-9 classes. Boisy Pitre had taken a job a few years earlier, and we met face to face at a CoCoFest then he started talking to me about working there. He had moved from development to training, and I joined. We'd fly to some location (Atlanta, Chicago, Princeton, Irvine, Orlando, etc.) with an OS-9/68K computer in a case, then check in at the hotel where the class would be. Course material and manual sets (back when it was six huge

binders) would have already been shipped, and we'd rent those old green/amber screen terminals from local suppliers. I'd spend the first night unpacking and hooking up everything in a conference room, then the next five days were spent with a class (usually no more than 8) going through starter, intermediate and advanced OS-9 topics. It was very hands on with lots of examples and coding. Great job. I got to see so many places thanks to it.

- What were they like to work for?

Amazing. And I hear it was better during the late 80s/early 90s before I joined. It was like you'd imagine Microsoft or Apple would have been. People wore whatever they wanted. Hours were flexible. We had a soda fountain with free pop. People were just incredibly nice, and there was cool stuff around every corner. I was shown a demo of D.A.V.I.D. (digital audio video interactive decoder) in 1995. This was a set-top box that could play digital video from some remote server. You could pause, rewind, fast forward video on demand. Or browse real estate listings or whatever. It was just a demo with a limited selection of video and choices, but the technology was all there. It even had games. It took almost 10 years before we got a real deployment here in Des Moines, though not running OS-9/DAVID. Too far ahead of it's time.

I also learned about something called Bluetooth and saw embedded Java long before it ever made it to cell phones or PDAs. There was so much cool stuff there.

- How long were you there?

Six days short of six years. I was hired in July of 1995 and laid off in July of 2001.

- Whatever happened to the company?

After going public shortly after I joined, Microware started losing money. It was strange because they were in four buildings, with several external rental properties. They had a huge three story building designed and built, including training rooms downstairs (so we didn't have to go to a nearby hotel for company-wide meetings). There was an exercise room (that contained video games for awhile, but never any exercise equipment). It was a great building, and with the investment by Motorola (they owned part of Microware) it seemed the sky was the limit. But, for whatever reason, Microware lost market share and was about to go broke. An Oregon-based company, RadiSys, purchased Microware in 2001. The two companies had been working together for some special Intel networking projects but it was a complete surprise the day they announced it to us.

Microware Communication Software Division existed for awhile as a subsidiary of RadiSys, then eventually became just part of RadiSys within the past few years. I actually returned as a contractor in 2003, and then was a full-time RadiSys employee eventually working on OS-9 again (support and maintenance). Most of the remaining Des Moines staff was working on other non-OS-9 related projects for the company, though. Most of the folks left have been there

for 15-20 years. It's a very elite senior staff and other than a few new folks brought in since I returned, I was the newbie in the building (with nearly ten years).

In January 2007 I left RadiSys, but the OS-9 spirit still resides there with new releases coming out a few times a year and some great new projects in the pipeline. Maybe we'll see a return of OS-9.

- What are your plans for the future now?

It looks like my OS-9 connected career path has finally ended, so it's off to look for something else in tech. Basically, I'm starting over with whatever I can find and learning all over again. I'll miss ERROR #216.

What other non-CoCo activities do you do now?

- Tell us about your keen interest in Hitchhikers Guide to the Galaxy?

I had a small TV set in my bedroom when I was in 7th grade. This was pretty rare back when most households still only had one set. In the TV Guide magazine I saw a listing for something called "Hitchhiker's Guide to the Galaxy" on the local PBS station. It sounded fun so I watched it. "It" was the BBC miniseries based on the original radio series (though I didn't know that then). The next day at school, in my junior high English class, I mimicked on of the lines a two headed character had said in the show. A guy in front of me turned around and said "you saw that?"

His name was Jimmy Jordan, and he forever changed my life. He introduced me to the book of Hitchhikers by loaning me his copies. He also introduced me to computers, and we'd meet at a local Radio Shack on weekends and type in programs on TRS-80 Model IIIs. He also introduced me to phone phreaking and a lot of other things. I can't think of any one person that had more impact on how my life turned out.

So that's how I learned of Hitchhikers and it literally "changed my life."

- Tell us about the other fairs you attend/organize.

As a kid, I got to take family trips to Disneyland and Walt Disney World. I also grew up going to Astroworld and Six Flags theme parks in Texas. I just loved all the themed entertainment, but there's not a lot of it in Iowa. A girl I was dating had an interest in renaissance stuff (I didn't; history was my worst subject in school) so we ended up attending a local renaissance festival that a Microware co-worker performed at. Somehow I made a connection between theme parks and renaissance festivals -- themed entertainment, just without the roller coasters.

Much like Sub-Etha, I wanted to find a way to make my hobby pay for itself. Road trips to remote events, costumes, food and lodging expense, etc. So I started selling roses, then started performing, then got more and more "well known" in the micro area thanks to websites and

photos I posted online (today over 30,000 renfest photos) that I was able to start helping out local events.

I originally did websites and show guides, then designed tickets and even helped create one small event from scratch with the folks that had the money to make it happen. Today I work a lot with the Des Moines Renaissance Faire which just opened in a permanent (all buildings) site in 2006. I designed the physical layout for the "village" for the "Festival Park" location. I can walk down a main street (inspired by Disneyland layout) and I know I placed those buildings there, and the large castle facade and jousting arena.

Officially, I'm just a helper, which means I don't have to take responsibility when something doesn't work, but I did entertainment scheduling, TV commercials and endless other things for the event (including many TV appearances all dressed up like a pirate for local news channels).

So, instead of just one or two CoCoFests a year, I often have years where I road trip to a dozen renaissance festivals (Iowa, Oklahoma, Illinois, South Dakota, Nebraska, Missouri, Kansas, etc.) and do interviews for a magazine I am editor of, or record audio and video for an internet pod cast.

And, unlike the CoCoFests, my renfest involvement has been quite profitable. It's also much fun with a lot of the similar "these guys are there for me" type friends... And, hanging out with a bunch of "rennies" in tights and pirate hats makes all of us CoCo guys look normal when we hang out together at the restaurant after a CoCoFest day :-)

You are in the envious position of owning one of the original, non-custom GIME chip prototype CoCo3's. Can you tell us how you obtained such a jewel?

When Microware was acquired in 2001, they stayed in the same three-story building that they had built a few years earlier during the "good times." Eventually, this building was sold and the new company would move the Des Moines facilities to a rented office space much more appropriate for the current staff size. This move required cleaning out the old building and "the morgue" ...a basement storage area.

The Morgue was an amazing place. There were so many OS-9 artifacts, from various types of MVME systems to CD-i players and development hardware. There was even an odd Japanese OS-9 "game machine" which I remember seeing before at, I think, the global OS-9 gathering in Chicago in 1993. I remember it having a space game where you navigated the file system by shooting at directories or something. And, of course, the Japanese version of "ERROR #216."

One item discovered was a set of cardboard boxes with Tandy/Radio Shack addresses on them. Peeking inside, I was puzzled at what this huge thing was but it had CoCo type ports on the back. Surely this couldn't be a prototype CoCo 3, could it? It turned out, it was, and a long-time IT/tech confirmed that. He had planned to save that artifact himself rather than throwing it out, and I was able to get it and show it off at the CoCoFest.

It does not contain a GIME chip, or even a prototype GIME chip like the machines Brother Jeremy had. Those machines were always referred to as "CoCo 3 Prototypes," so I guess what I have is a prototype to the prototype... Instead of a GIME, it had a large array of chips wired together that would later be turned in to a custom IC. It is hoped this will eventually allow us to decode everything the GIME did, and see if the early prototype did indeed contain extra graphics modes like the rumored 256-color mode.

If you could go back in time and change just one small thing in the CoCo timeline, what would you choose?

I think a key turning point in my CoCo life was when The Rainbow magazine decided they would not cover any of the "next generation" machines like the MM/1, Tomcat, TC-9, and Delmar systems. I understand why they did that since, after all, they were not CoCos at all -- just machines designed by people familiar with the CoCo and based on an OS the CoCo could optionally run. But, they would have made a natural hobbyist progression, taking us to a more modern hardware platform while retaining some of the spirit of what we loved about the CoCo. It was because of this lack of coverage that I decided to drop my subscription to Rainbow. That was a mistake. I should have supported them to the end, and maybe we could have kept things going a bit longer.

So that's it. Sure, I wish a CoCo 4 was actually released, and we think it was planned. When I worked for Radio Shack, our area manager swore that she got to see a CoCo in a Tandy 1000 HX type case with a built in disk drive. To this day, there is still little known about what (if anything) Tandy was trying to do with the CoCo. Even in the final years of production it was still a top selling computer for the company, in number of units sold over the holidays. There were also plenty of signals, like when the FD-502 disk drive was discontinued completely. We were told that users would be able to buy the external HX drive and hook it up to a CoCo with some new interface, which would have given then CoCo an official 3.5" floppy drive. It seems there were things going on at Tandy Towers before the End, and I would have liked to see support continue a few more years just to see what they had planned.

An Introduction to



By Taylor Software

This tutorial was written to give you the information you need to get your first assembler file built and put onto an emulator pseudo-floppy disk. It will be centred on using the Tandy Colour Computer emulation, since that is what I use. I'm sure it will be equally applicable to other hosts.

Created by Bob Devries

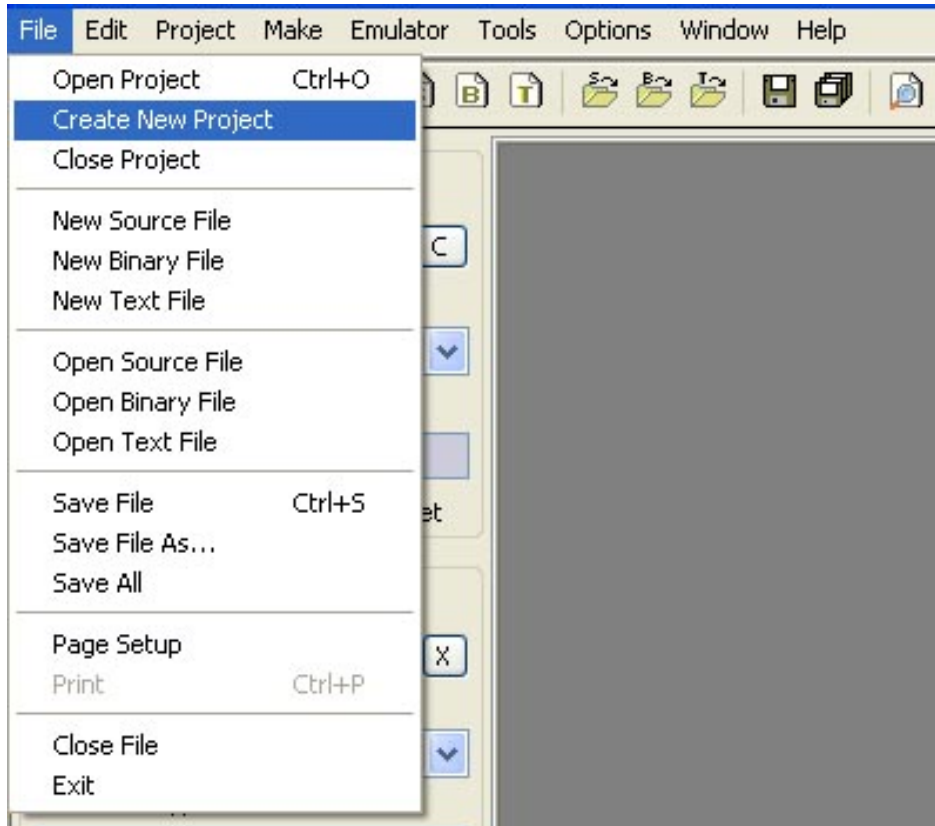
Bob Devries

To start off, the Rainbow IDE Installer does not create an icon on the desktop. To make the programme easy to use, do this now. For those who don't know how; click **Start->All Programs** and highlight the Rainbow IDE executable. Then right-click it, and select **Send To->Desktop (create shortcut)**. Now you're ready to work.

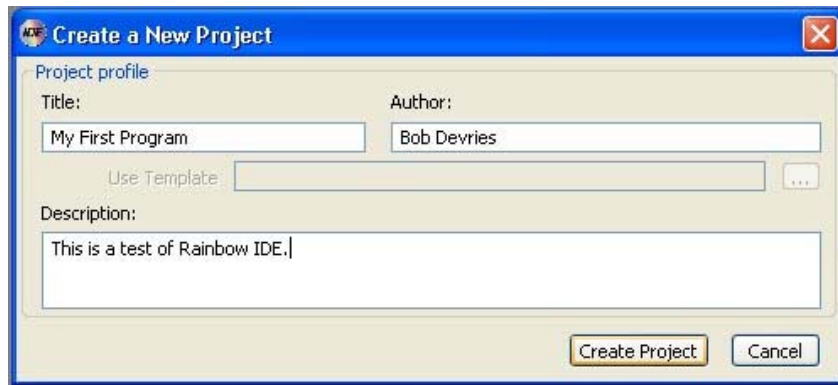
Double-click the **Rainbow IDE** icon to launch the programme.

I am going to guide you through creating a Colour Computer binary file written in assembly language, and written to a pseudo disk for the MESS emulator. Don't worry; the assembler will be very simple; I'm not here to teach you assembler.

The **Rainbow IDE** works on projects, each with a unique name. I will call ours **My First Programme**. To create a new project, click on the **File** pull-down menu, and select **Create New Project**:



You will be presented with a dialog box into which you can type the name of your project, like this:



Type in the title, description and the author as I have shown above. The description and author are optional. When ready, click the **Create Project** button at the bottom right of the dialog box.

The next thing to do is to add an assembler source file to this project. This is done easily using a button on the left end of the toolbar marked with an **S**. The picture below will show you where it is:

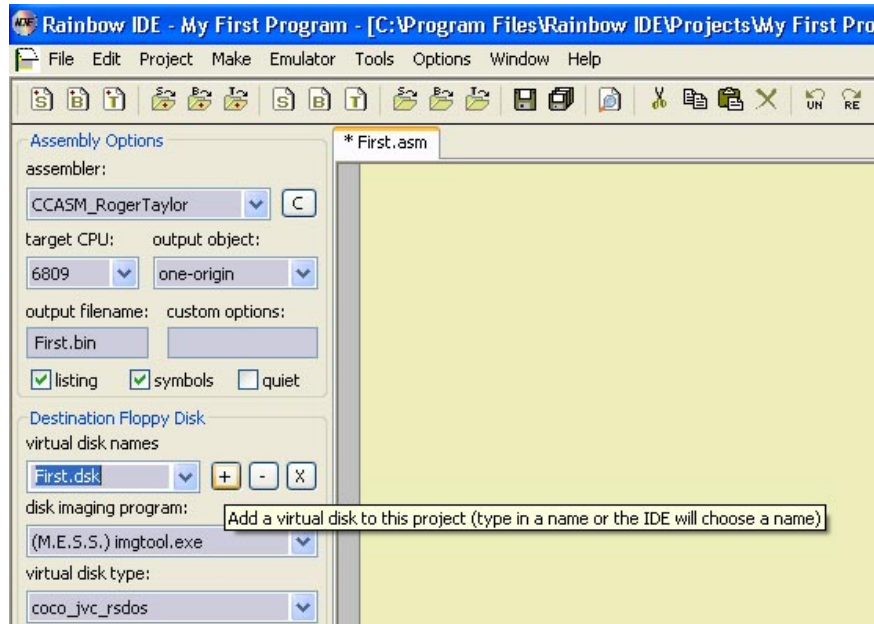


This will present a dialog box asking for the name of the source file. If you are working with assembler files, the extension may be **.a**, **.asm**, **.src**, **.as9**. When that's done, you need to choose the assembler type from the drop-down list; the target CPU; the output object, and output file name. These drop-down menus are seen in the picture above, below the tooltip (the yellow box). For the Tandy Colour Computer, you choose either 6809 or 6309, and the object can be any of a number of choices. I have chosen one-origins for a simple binary file.

The output filename will be the name that is used for the file on the pseudo disk, and must comply with the host's naming conventions. For the Tandy Colour Computer, that means eight (8) characters, with three (3) for the extension, which is usually **.BIN**.

The next part is a bit tricky, and can trip you up if not done right.

You need to choose a name for the pseudo disk. Rainbow IDE will choose its own if you don't supply it, but whether you do or not, you must click on the button with the + on it. See the picture below.



If you do not type a name for the pseudo disk, Rainbow IDE will use **disk1.dsk**. These pseudo disk files will be created in a folder in the directory with the name of your project in the tree of folders of Rainbow IDE in the Program Files folder.

Now you are ready to type your code into the text window. It has the source file name at the top with an asterisk next to it. Click on the window to activate it, and start typing. You can also cut and paste from another editor or file viewer.

Once you have completed typing the source code you should save it using the toolbar disk icon for saving the current file, or click on the multiple disk icon for saving all files. Now, you can **build** the project by clicking on the **build** button on the right of the screen (under the pesky **Please Register** window on unregistered versions). This will create a new screen which will show the progress of the build, and also errors if the should occur.

Finally, when that is all done, your new programme can be run in the MESS emulator. Before this can happen you must configure MESS correctly, firstly in the **Options->M.E.S.S. Options** pull-down menu. There you can select the path to your copy of the MESS emulator. Also, the **launch** button must be clicked for MESS to start.

The MESS emulator will be started with your pseudo disk loaded and you can load the binary file, and execute it. On the Tandy Colour Computer, that is done using **LOADM** and **EXEC**.

HAPPY HOLIDAYS!

*Just in time for the
gift giving season!*



**Hot CoCo Gifts
from
The CoCo Hut
Gift Shop**

New items coming soon...

6809 www.cafepress.com/cocohutshop

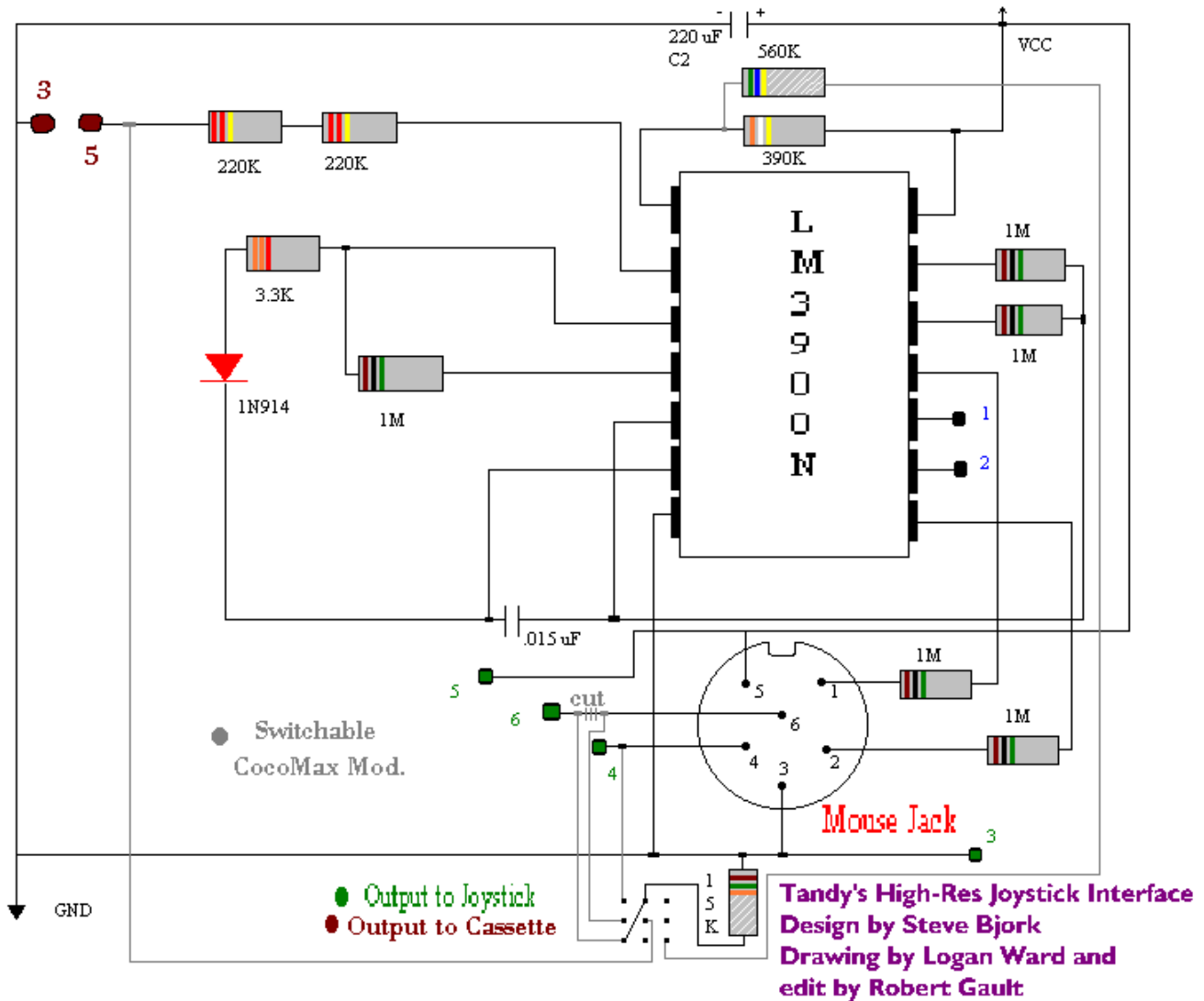
Programming for the Tandy High Resolution Joystick Interface

by
Robert Gault

One of the limitations of the Color Computer is the Digital to Analog Converter (DAC). This device is used to detect joystick positions and convert them to usable numbers. It is also used to generate sound and signals for recording on tape. The problem with the device is the limit of 64 discrete levels. Why this is a problem when generating audio output would take several articles. This discussion will be limited to joystick sampling.

The most frequent use of joysticks is to move a cursor around a graphics screen. Even the Coco1 in PMODE4 mode has a 256x192 screen where both axes far exceed the maximum DAC value of 64. The Coco3 with a 640x192 or higher resolution further exceeds the DAC capacity. So, what does a programmer normally do? With a Coco1, the joystick value must be multiplied by 4 ($4 \times 64 = 256$) or 3 ($3 \times 64 = 192$). The Coco3 would need $10 \times 64 = 640$ for the horizontal dimension. This means that it is not possible to test every pixel on the graphics screen on any Coco. On a Coco3 with a width of 640, a pointer can move to only one in every ten positions resulting in very jerky movement.

To overcome this limitation, Tandy sold a high resolution joystick interface 26-3028. A slightly different design was used for the Coco Max III graphics program. The Tandy unit was reverse engineered by Logan Ward and the circuit diagram follows.



The device generates a voltage ramp which charges a capacitor. The time taken to charge the capacitor to the joystick voltage, equivalent to its position, can be measured by a software routine. This can increase the resolution of the joystick so that every position on a 640x192 screen becomes accessible.

The critical part of the design is permitting the capacitor to completely discharge so a joystick reading of zero is possible and that the rate of charging permits a reading of 639 to be made. The Tandy design works, but the CocoMax III modification as shown above charges too rapidly so the full 640 screen can't be accessed. Since the CocoMax program did not need to access a full 640 screen, the design was adequate for that program.

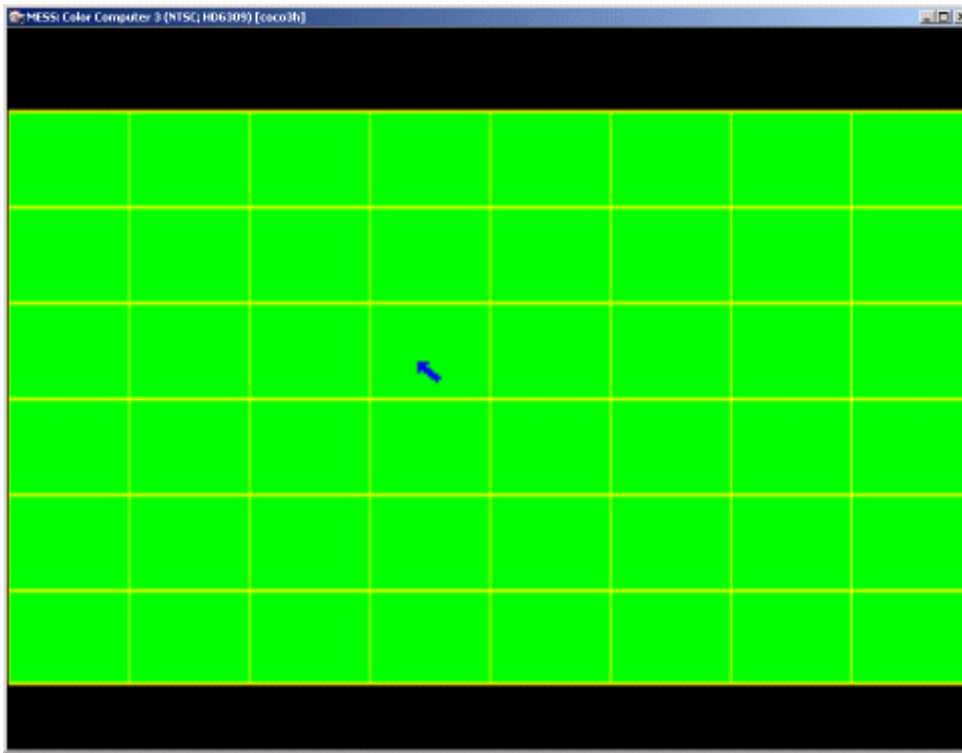
There is not much more that needs to be said about the device and the following programs speak for themselves with line by line documentation. The assembly source code is

not the only way to read the interface and is intended as a teaching device rather than an optimized program. The demo is written for the Coco3 and may not work with all emulators. It will work with MESS v.104.

Change line 20 in the Basic program to load either the Tandy or CocoMax binary. The rest of the program will be unchanged. The Coco must run at 2MHz for the program to work to be fast enough to work.

```
10 ' DEMO ROUTINE BY ROBERT GAULT TO INDICATE HOW HIGH RES
ADAPTOR WORKS
20 LOADM"HRTANDY",&H7000:' USE WHATEVER NAME YOU SAVED THE ML
ROUTINE UNDER
22 POKE&HFFD9,0:' MUST USE FAST 2MHz MODE
30 RGB:' OR CMP OR SET PALETTE AS DESIRED BUT MAKE COLORS 0-3
DIFFERENT
40 HSCREEN4:' ALSO TRY HSCREEN 1 OR 2; FOR 3 CHANGE ARROW COLOR
50 HBUFF1,80:HBUFF2,80:' LARGE ENOUGH FOR HSCREEN2 USE
60 H$="NM+11,+4":' PART OF DRAW STRING CALLED SEVERAL TIMES
70 BF=1:GOSUB200:GOSUB230:BF=2:GOSUB200:GOSUB210:' BF SELECTS
BUFFER #
80 JS=&H7004:XVAL=&H7000:YVAL=&H7002
90 DEF FNA(X)=PEEK(X)*256+PEEK(X+1):' CONVERTS A 2 BYTE NUMBER
TO BASIC INTEGER
100 FORX=0TO640STEP80:HLINE(X,0)-(X,191),PSET:NEXT:' DRAW GRID
LINES
110 FORY=0TO192STEP32:HLINE(0,Y)-(639,Y),PSET:NEXT
120 X=1000:Y=X:' SETUP INITIAL CONDITIONS
130 '
140 FORT=0TO1STEP0:EXEC JS:' INFINITE LOOP: READ JOYSTICK
150 A=FNA(XVAL):B=.3*FNA(YVAL):' RECOVER X & Y VALUES
160 IFX>A-.5 AND X<A+.5 AND Y>B-.5 AND Y<B+.5 THEN 140:' TEST
FOR NO CHANGE
170 IF Z<>0 THEN GOSUB210 ELSE Z=1:' DON'T RESTORE SCREEN ON
FIRST PASS
180 X=A:Y=B:BF=1:GOSUB200:GOSUB220:NEXT T
190 '
200 HGET(X,Y)-(X+16,Y+7),BF:RETURN:' USED FOR BOTH BUFFER 1 & 2
210 HPUT(X,Y)-(X+16,Y+7),1:RETURN:' ORIGINAL SCREEN VALUES
220 HPUT(X,Y)-(X+16,Y+7),2,OR:RETURN:' ARROW ICON
230 HDRAW"C2;BM 0,0;NR7;D1;NR5;D1;NR3;D1;NR1;BM+2,-1;XH$;BM+0,-
1;XH$;BM+2,+0;XH$;C1":RETURN:' CREATE ARROW ICON
240 '
250 This program simulates the cursor action present in OS-9
MultiVue. In
260 most cases, frequent calls to a cursor update subroutine
should be OK
```


270 for BASIC programs. If greater speed is required, minor modifications
 280 to the ml. program could incorporate the routine into the
 IRQ vector.
 290 This would make joystick polling independent of the BASIC
 program.
 300 For the gain to be significant, the ml. routine would also
 have to draw
 310 the arrow icon; not that easy a task in the context of an
 RSDOS system.



```

00010 * TANDY TYPE HIGH RESOLUTION JOYSTICK ROUTINE
00020 * (c) NOV. 1992 BY ROBERT GAULT
00030
00040 * READS RIGHT JOYSTICK AND STORES X & Y VALUES AT OFFSET
00050 * 0 & 2; ROUTINE STARTS AT 4
00060
00070 * COCO3 FAST MODE ONLY; COMPENSATES FOR 320/640 FORMAT
00080 * SHOULD BE OFFSET LOADED TO DESIRED ADDRESS, PIC CODE
00090
00100     ORG     0
00110 XVAL  RMB   2
00120 YVAL  RMB   2
00130 START PSHS CC
00140     SYNC           REDUCES JITTER
00150     ORCC  #$50    TURN OFF INTERRUPTS
00160     LDA   $FF23  SOUND CONTROL; SAVE AND TURN OFF
00170     STA   <SOUND,PCR

```

```

00180      ANDA  #.NOT.8          NO SOUND
00190      STA  $FF23
00200      LDA  $FF01 MUX VALUES; SAVE AND SET FOR X JSTICK
00210      STA  <PIA0A,PCR
00220      ANDA  #.NOT.8          MUX=0
00230      STA  $FF01
00240      LDA  $FF03
00250      STA  <PIA0B,PCR
00260      ANDA  #.NOT.8          MUX=0
00270      STA  $FF03 MUX IS NOW 0,0
00280      LDX  #$FF00          SET FOR INDIRECT INSTRUCTION
00290      BSR  READJS          READ JOY STICK VALUE; X
00300      STD  <XVAL,PCR      SAVE THE RAW ANSWER
00310      LDA  <PIA0A,PCR      SET MUX FOR Y JSTICK
00320      ORA  #8             MUX=1
00330      STA  $FF01          MUX NOW IS 0,1
00340      BSR  READJS          READ JOYSTICK; Y
00350      STD  <YVAL,PCR      SAVE ANSWER
00360      LDD  <PIA0A,PCR      RESTORE ORIGINAL MUX VALUES
00370      STA  $FF01
00380      STB  $FF03
00390      LDA  <SOUND,PCR      RESTORE ORIGINAL SOUND VALUES
00400      STA  $FF23
00410      LDA  $E6           HRES GRAPHICS MODE INDICATOR
00420      CMPA #3           HAVE WE A 320 OR 640 SCREEN
00430      BHS  E0
00440      LDD  <XVAL,PCR      IF 320; DIVIDE X VALUE BY 2
00450      LSRA
00460      RORB
00470      STD  <XVAL,PCR
00480 E0    PULS  CC,PC RETURN TO CALLING PROGRAM
00490
00500 SOUND RMB  1
00510 PIA0A RMB  1
00520 PIA0B RMB  1
00530
00540 READJS      LDD  #$FE8C          REG.A=DAC; REG.B=DISCHARGE COUNT
00550      STA  $FF20          TURN OFF RAMP IF ON
00560 A0    DECB          WAIT FOR FULL DISCHARGE
00570      BNE  A0
00580      LDD  #809          MAX COUNT
00590      PSHS  A
00600      LDA  #2           RAMP TRIGGER
00610      STA  $FF20          START RAMP
00620 B0    LDA  ,X          TEST DAC FOR MATCH
00630      BMI  C0           EXIT IF MATCH
00640      SUBB  #1          LSB COUNT
00650      BHS  B0
00660      DEC  ,S          MSB COUNT
00670      BPL  B0
00680 C0    LDA  #$FE
00690      STA  $FF20          STOP RAMP
00700      PULS  A
00710      TSTA
00720      BMI  OVFLOW
00730      PSHS  D
00740      LDD  #639

```

```

00750      SUBD   ,S++
00760      BCC   D0
00770      LDD   #0          PREVENT UNDERFLOW
00780 D0     RTS
00790 OVFLOW      LDD   #639          PREVENT OVERFLOW
00800      RTS
00810
00820      END   START
      END   START
END   START

00810
00820      END   START

00010 * COCOMAX TYPE HIGH RESOLUTION JOYSTICK ROUTINE
00020 * (c) NOV. 1992 BY ROBERT GAULT
00030
00040 * READS RIGHT JOYSTICK AND STORES X & Y VALUES AT OFFSET
00050 * 0 & 2; ROUTINE STARTS AT 4
00060
00070 * COCO3 FAST MODE ONLY; COMPENSATES FOR 320/640 FORMAT
00080 * SHOULD BE OFFSET LOADED TO DESIRED ADDRESS, PIC CODE
00090
00100      ORG   0
00110 XVAL  RMB   2
00120 YVAL  RMB   2
00130 START PSHS CC
00140      SYNC          REDUCES JITTER
00150 ORCC  #$50  TURN OFF INTERRUPTS
00160 LDA  $FF23 SOUND CONTROL; SAVE AND TURN OFF
00170 STA  <SOUND,PCR
00180 ANDA #.NOT.8          NO SOUND
00190 STA  $FF23
00200 LDA  $FF01 MUX VALUES; SAVE AND SET FOR X JSTICK
00210 STA  <PIA0A,PCR
00211 ANDA #.NOT.4
00212 STA  $FF01
00213 LDB  #4
00214 STB  $FF00
00215 ORA  #4
00216 STA  $FF01
00220 ANDA #.NOT.8          MUX=0
00230 STA  $FF01
00240 LDA  $FF03
00250 STA  <PIA0B,PCR
00260 ANDA #.NOT.8          MUX=0
00270 STA  $FF03 MUX IS NOW 0,0
00280 LDX  #$FF00          SET FOR INDIRECT INSTRUCTION
00290 BSR  READJS          READ JOY STICK VALUE; X
00300 STD  <XVAL,PCR      SAVE THE RAW ANSWER
00310 LDA  <PIA0A,PCR    SET MUX FOR Y JSTICK
00320 ORA  #8            MUX=1
00330 STA  $FF01          MUX NOW IS 0,1
00340 BSR  READJS          READ JOYSTICK; Y
00350 STD  <YVAL,PCR      SAVE ANSWER
00360 LDD  <PIA0A,PCR    RESTORE ORIGINAL MUX VALUES

```

```

00361     ANDA  #.NOT.4
00362     STA  $FF01
00363     CLR  $FF00
00364     ORA  #4
00370     STA  $FF01
00380     STB  $FF03
00390     LDA  <SOUND,PCR  RESTORE ORIGINAL SOUND VALUES
00400     STA  $FF23
00410     LDA  $E6      HRES GRAPHICS MODE INDICATOR
00420     CMPA #3      HAVE WE A 320 OR 640 SCREEN
00430     BHS  E0
00440     LDD  <XVAL,PCR  IF 320; DIVIDE X VALUE BY 2
00450     LSRA
00460     RORB
00470     STD  <XVAL,PCR
00480 E0    PULS  CC,PC RETURN TO CALLING PROGRAM
00490
00500 SOUND RMB  1
00510 PIA0A RMB  1
00520 PIA0B RMB  1
00530
00540 READJS     LDD  #$478      REG.A=DAC; REG.B=DISCHARGE COUNT
00550     STA  $FF00      TURN OFF RAMP IF ON
00560 A0    DECB      WAIT FOR FULL DISCHARGE
00570     BNE  A0
00580     LDD  #639      MAX COUNT; 230
00590     PSHS A
00600     LDA  #0      RAMP TRIGGER
00610     STA  ,X      START RAMP
00620 B0    LDA  ,X      TEST DAC FOR MATCH
00630     BMI  C0      EXIT IF MATCH
00640     SUBB #1      LSB COUNT
00650     BHS  B0
00660     DEC  ,S      MSB COUNT
00670     BPL  B0
00680 C0    LDA  #4
00690     STA  $FF00      STOP RAMP
00700     PULS A
00710     TSTA
00720     BMI  OVFLOW
00730     PSHS D
00740     LDD  #639
00750     SUBD ,S++
00760     BCC  D0
00770     LDD  #0      PREVENT UNDERFLOW
00780 D0    RTS
00790 OVFLOW     LDD  #639      PREVENT OVERFLOW
00800     RTS
00810
00820     END  START

```

Color Computer Tray By Militant Buddah

It all started while playing with Color Computer software on M.E.S.S. at work. It was fun, but it was missing something. Something that kept the experience from being more authentic. Then it hit me. It's more fun actually typing on a Color Computer. Well, dragging a Color Computer and the rest of the equipment to work wasn't exactly feasible.

I use a docking station for my laptop at work and use a real keyboard attached to it, which got my gear turning.

I had an old dead 64k Color Computer that died years ago in a lighting strike. I stripped out the guts. You should have seen the scorching on the motherboard! I bought a tiny keyboard from Fry's, along with a USB hub to put inside, which is mounted in the cartridge slot. I had to use a Rotary tool to cut out the tray some to make the keyboard fit and made a mount for the keyboard and the USB2.0 hub. I use it at work as my docking station keyboard now. It gets a lot of attention. My manager thought I actually had an old computer on my desk when he stopped by when I was eating lunch in my cube and I had M.E.S.S. fired up. He used to have one and sat down for a while and played with it. It definitely helps makes the experience seem more authentic.

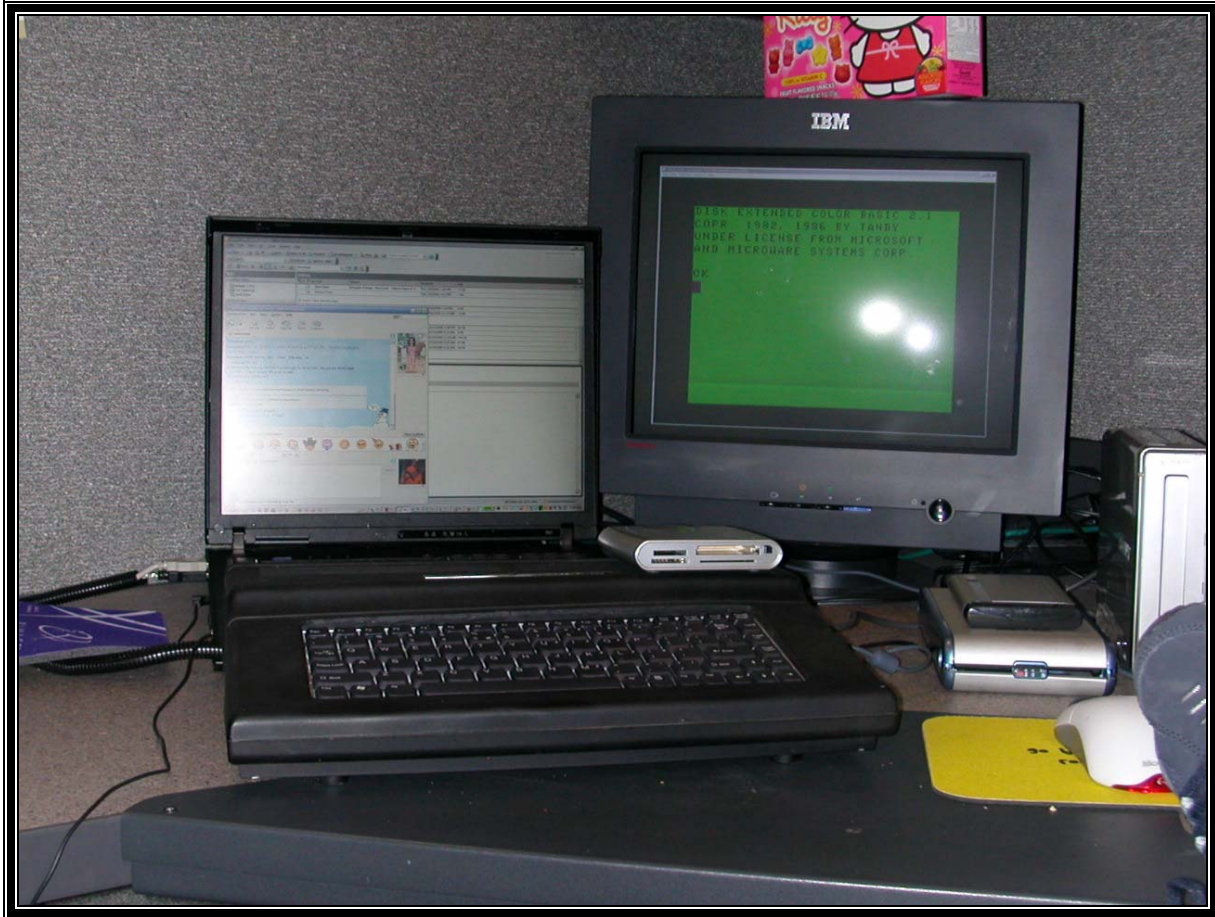


Nice, but missing something

It was cool and I was enjoying it quite a bit. But the contrast between the keyboard and the case bothered me some. It lacked "ZAZZ". Well, I was never crazy about the beige seem that 90 percent of all computers seem to use. I liked the look of the silver COCO's much better. But silver doesn't paint well at all, so I thought, "Why not black?" I like the look of black computers like the modern IBM workstations, so why not for the COCO? I like the look much better now. I added a media card reader. It can't be seen in the picture, but the COCO label on the top reflects the memory of 1 gig in kilo bites.

So now I can do my work and play using dual monitor setup on my Thinkpad. The two blend in together quite well, don't they?





My next project is to take a Model III or IV (which to me, is the coolest looking computer ever) and convert it into a real computer. Soooo NCC-1701 original Star Trek looking. I already have tracked down a 13 inch CRT and an old P4 1.6 with 2 gigs of memory. I plan to use it as my FTP server. If I get that done, maybe Mary will post that.

“Maybe It’s a Library. Maybe It’s a Script. But It’s Definitely a Time-saver.”

Article by Richard Kelly

ECB creations that use LINE, DRAW, CIRCLE, etc. are quite common. When programming graphical adventures and room-based games though, the coding gets very repetitious in a hurry.

The way ECB works, words like CIRCLE, PAINT, PSET, and LINE are stored as one byte when they’re typed in or saved. While this is more efficient for the computer, it doesn’t do squat for the programmer. After all, he’s the one who has to type in the same entire words over and over and over again. *And* he has to make sure that the spelling of each word is correct, lest he shot by ECB with an “?SN ERROR”.

Even worse, he even has to keep typing in all those special symbols along with the commands, and do all of those correctly as well. The one command LINE/B works like this:

>LINE(10,20)-(30,40),PSET,B

It's be a lot less to type in if we could do something like this instead:

>L,10,20,30,40,B

But ECB won't let us to skip steps with its commands, no matter how many times we've used that same command before in the very same program.

Certainly, there's **got** to be a less time-consuming, less tedious way of doing this. Sure enough, there is. Check this program out, which draws a crude picture of a house:

```
>1 PMODE0:PCLEAR4:PMODE4:SCREEN1,1
>8000 READA$:A=VAL(A$):Z=A<0:ONZ+2GOTO9999:
ONA+1GOTO8001,8002,8100,8003:READX,Y:
ONA-3GOSUB8004,8005,8006,8007,8008,8009,8010:GOTO8020
>8001 DRAWA$:GOTO8020
>8002 READA$:PLAYA$:GOTO8020
>8003 READX:PCLSX:GOTO8020
>8004 PMODEX,Y:RETURN
>8005 READA,B:LINE(X,Y)-(A,B),PSET,B:RETURN
>8006 READA,B:LINE(X,Y)-(A,B),PSET,BF:RETURN
>8007 READZ:CIRCLE(X,Y),Z:RETURN
>8008 READZ:PAINT(X,Y),Z,Z:RETURN
>8009 READZ:PSET(X,Y,Z):RETURN
>8010 PCOPYX TOY:RETURN
>8020 ' PLACE ANY PAUSES BETWEEN STEPS IN THIS LINE.
>8030 GOTO8000
>8100 READX:ONX+1GOTO8101,8102
>8101 FORT=0TO49:X=RND(256)-1:Y=RND(144)-1:
Z=RND(3)+1:PSET(X,Y,Z):NEXT:GOTO8020
>8102 X=110:Y=112:Z=146:FORT=-1TO0:LINE(X,Y)-(Z,Y),PSET:
Y=Y-1:X=X+1:Z=Z-1:T=Z>X:NEXT:GOTO8020
>8999 ' 0=DRAW;1=PLAY;2=SPECIAL FUNCTION;3=PCLS;4=PMODE;
5=LINE,B;6=LINE,BF;7=CIRCLE; 8=PAINT;9=PSET;10=PCOPY
>9000 DATA 4,1,1, 3,1, 4,1,3, 3,2, 10,2,3, 4,3,1, 2,0, 4,4,1
>9010 DATA C1, 6,108,113,148,143, 2,1, 4,3,1, C3, 6,140,137,144,143, 4,4,1
>9020 DATA "BM104,117;C1;E24F24;BR2;H26G26;BL2;E28F28;BR2;H30G30"
>9030 DATA9,240,16,0, 7,240,16,10, 8,240,16,5, 9,240,6,0, 9,250,16,0, 9,240,26,0,
9,230,16,0
>9040 DATA 1,"O3T3L8;CD;L2;E;L4;FA;L2;G;L4;EG;L4.F;L8;E;L4;FD;L2;EP4"
>9050 DATA 1,"L8;CD;L2;E;L4;FA;L2;G;L4;EG;L4.F;L8;E;L4;FD;L1;C"
>9090 DATA -1
>9999 END:RUN
```

The code above demonstrates a script (or library, if you prefer to call it that) that makes calling common commands in ECB shorter and simpler. The script works by reading from

DATA statements, then using the first number it reads to call the function you want, reading any other numbers its needs to know to perform the function. If the first number it reads isn't a number at all, the program assumes it's a string and performs the DRAW function. As such, the COLOR command isn't necessary in the script; you can just enter in "C1", "C2", "C3" or "C4" without quotes to set the color. Here's a quick summary of the script and how to use it.

Function: DRAW

To call, use: the string you want the DRAW command to use.

Example: DATA "BM0,0;M128,96" will do the command DRAW "BM0,0;M128,96".

Function: PLAY

To call, use: 1, then the string.

Example: DATA 1,"T2L8O3;C;L8;C" will do the command PLAY "T2L8O3;C;L8;C".

Function: PCLS

To call, use: 3, then the number to clear the screen to.

Example: DATA 3,1 will do the command PCLS 1.

Function: PMODE

To call, use: 4, the first number, a comma, then the second number.

Example: DATA 4,0,1 will do the command PMODE 0,1.

Function: LINE/B

To call, use: 5, then the four numbers required, each separated by commas.

Example: DATA 5,10,20,30,40 will do the command LINE(10,20)-(30,40),PSET,B.

Function: LINE/BF

To call, use: 6, then the four numbers required, each separated by commas.

Example: DATA 6,10,20,30,40 will do the command LINE(10,20)-(30,40),PSET,BF.

Function: CIRCLE

To call, use: 7, then the first three numbers the CIRCLE command uses, separated by commas.

Example: DATA 7,128,96,10 will do the command CIRCLE(128,96),10.

Function: PAINT

To call, use: 8, then the first three numbers PAINT uses, separated by commas. The script automatically selects the color of the "paint" as the color that PAINT won't overlap. If you want to select the "anti-overlap color" yourself in the script, you can always change the program code to read an extra number. It's located in line 8008 of the code.

Example: DATA 8,128,96,1 will do the command PAINT(128,96),1,1.

Function: PSET

To call, use: 9, then the three numbers the PSET function uses, separated by commas.

Example: DATA 9,128,96,1 will do the command PSET(128,96,1).

Function: PCOPY

To call, use: 10, then the two numbers PCOPY uses, separated by a comma.
Example: DATA 10,1,2 will do the command PCOPY 1 TO 2.

Function: End the script routine.

To call, use any number from -1 or lower. This allows you to “mark” each routine by a number, even if it is a negative one. This is great for adventure games!

Example: DATA -1 will end the script routine. So will DATA -2 and DATA -3, and so on.

Now, the part of the script called by “2” requires some explanation. This is a completely user-defined function where you can call FOR/NEXT commands and other things that are tedious or inefficient to call from the script. When you enter in “DATA 2,0” to the script, it goes to line 8101 of the code, and draws several randomly colored “stars” onto the screen. When you enter in “DATA 2,1” to the script, it goes to line 8102 of the code, and draws LINES using a FOR/NEXT loop. You don’t want some of these commands drawn before the script is started, nor after the script routine ends, or the “stars” will be partly erased, or will overlap what they’re not supposed to (like the moon or the front of the house). You want the routines executed sometime in the middle, and that’s what the “2” function is for. Just change lines 8100-8199 in the code to whatever you want to fit your program’s needs.

The script also helps solve common problems where the user puts the wrong coordinates in the code somewhere, but he isn’t sure where. Just modify line 8010 of the program with a delay or INKEY\$ routine so you can track down right where the problem is, and fix it. This is a heckuva lot better than inserting delay commands in every single graphics statement you’ve written! Just remember that the code will pause when doing PMODE and C1, C2, C3, and C4 commands as well, so sometimes it will appear to pause twice with no visible changes, unless you’ve programmed the code to be “smarter” than this.

Changing the code to run CoCo 3 graphics modes is easy, to. Just change lines 1 and lines 8001-8010 and you’re all set. For the program example I’ve given you, you’d also have to change lines 8101 and 8102 before the program will run. Lastly, there’s no point to trying to use the equivalent of PCLS, PCOPY, or PMODE in HSCREEN, so I would change them to HPRINT, ATTR, and PALETTE.

The above sample program could use some touching up. I invite you to put in some extra instructions to the code by inserting a LINE 9005, and fix things so the stars don’t show through the roof of the house, nor “distort” the shape of the moon. Here’s a hint - The quickest way to do this is with one DRAW command and one CIRCLE command. Have fun! ☺

X-FILE: A file manager for Disk Basic.

By diego



Even since I had only a cassette system, I had a small program that loaded some patches and settings I liked to use on my CoCo. When I've got my first disk system, it was just natural to do the same.

It all started with a program that did a few pokes, some "paletteing" and then, just listed a DIR. Then, it started growing.

I keep this program in all my disks, saved with a short name ("XF" right now, "OS" was another option). That way, when I insert a disk in the drive, instead of typing DIR, that can be quite useless if you have more files than screen lines, I just do a RUN "XF". Takes a bit longer, but it's usually worth it.

This version, 2.3 supports any Color Computer with 1 to 4 disk drives. I have a CoCo 3 only version, that is smaller and takes only 2 granules of disk space. It will be soon available in my website,

<http://coco.sclaudia.net>. Anyway, is not that hard to delete the parts that do not match your system, but remember that not to much was done to optimize the display to the 32x16 screen (The display is limited to 24 files...)

LOADING THE PROGRAM:

Just load it/ run it from your current drive.

If you have 1 or 2 double sided disks, the program will map the second side of drive 0 to drive 2 and drive 1 to drive 3.

If you have more than 2 drives, you should remove the line 210 POKE 55455,65:POKE 55456,66

If you are using a CoCo 1 or 2, the display will be in the 32x16 screen.

In a CoCo 3, the 80x24 is default, but the only change needed to use the 40x24 screen is change the WIDTH command in line 40. The display format will adjust to the screen size with no need for any other change.

USING THE PROGRAM:

Most file-managers expect that you first select a file, and then decide what to do with it.

X-File works the other way. First, you choose the command, then the file.

The available commands are:

<L>oad

Loads a file into memory.

The program will recognize the file format and use the appropriated command. If you attempt to load a data file, an error message will be displayed.

Since many ML programs will autorun and expect a 32x16 screen, the program switches to that mode before LOADM.

Note that loading a BASIC program will overwrite X-File in memory.

<C>opy

Creates a copy of a file.

After selecting the file to be copied, the program asks for a destination name.

You can enter any valid file name to create a copy, and must include any desired extension.

NEWFILE.BAS will make a copy of the file in the same drive.

NEWFILE.BAS:1 will make the copy in the disk in drive 1.

You can cancel the copy command by entering a blank destination name

<M>ove

Moves a file.

Same as COPY, but will delete the original file.

You can cancel the move command by entering a blank destination name.

<D>elete

Deletes a file.

The program will ask for confirmation before proceeding.

<R>ename

Changes the name of a file

You can cancel the rename command by entering a blank destination name

<U>ndelete

Recovers a deleted file from the disk.

You have to give a first letter for the name of the file to be recovered. This can be any valid character.

This option only recovers the first granule of the file. If the original file was bigger, it might still recover some useful information.

I do recommend to copy the recovered file to another disk and re-erase the original, just in case that the recovery process might affect the disk contents.

<I>nstall

Saves the X-File program to the disk in the current drive, using the name "XF.BAS"

<0><1><2><3>

Changes the current drive and lists the contents of the disk in that drive.

SELECTING A FILE

You can use the cursor and ENTER keys to select a file.

Also, you can just start typing the file name. The cursor will move to the first file whose name matches the letters you have typed so far.

Always use ENTER to select the file.

The CLEAR key cancels the file selection and brings you back to the main screen.

I hope you enjoy using this program as much as I did writing it. I appreciate any feedback to diegoba@adinet.com.uy

```
10 REM XFILE V2.3 NOV 19, 2006
20 CLEAR 2000:DIM EX(72),FI$(72),WF$(72),DF$(72)
30 IF PEEK(&HFFFE)<>140 THEN 150 'CHECK IF COCO 1/2-3
40 WIDTH 80
50 C3=1
60 POKE 65497,0
70 ON BRK GOTO 10000
```

```

80 ON ERR GOTO 9000
90 POKE &HE649,&H10:PRINT"Palette Fixed"
100 PALETTE RGB
110 PRINT"Disk head step rate set at 6ms"
120 PRINT"I/O ERROR retries: 2"
130 PRINT"Access to second side of disks, enabled as drives 2 and 3"
140 GOTO 190
150 POKE 65495,0
160 PRINT"DISK HEAD STEP RATE SET AT 6MS"
170 PRINT"I/O ERROR RETRIES: 2"
180 PRINT"ACCESS TO SECOND SIDE OF DISKS, ENABLED AS DRIVES 2 AND 3"
190 POKE 55232,0:POKE 55318,20
200 POKE 55138,2
210 POKE 55455,65:POKE 55456,66
220 SW=PEEK(&H00E7) ' SCREEN WIDTH
230 IF C3=0 THEN SW=0
240 IF SW=0 THEN CP=1055 ELSE CP=442369
250 IF SW=0 THEN BL$=STRING$(31," ") ELSE IF SW=1 THEN BL$=STRING$(39,"
") ELSE BL$=STRING$(79," ")
490 FOR ML=0 TO 1 STEP 0:SOUND 255,5:SOUND 100,1
500 D=PEEK(&H95A) ' CURRENT DRIVE
510 GOSUB 1010 ' LIST FILES
520 IF SW=0 THEN PRINT@480,"LOAD COPY MOVE DELETE
RENAME";PRINT@448,"DRIVE 0/1/2/3 UNDELETE INSTALL";
530 IF SW=1 THEN LOCATE 0,22:PRINT "Drive <0><1><2><3> <U>ndelete
<I>nstall";LOCATE 0,23:PRINT"<L>oad <C>opy <M>ove <D>elete <R>ename";
540 IF SW=2 THEN LOCATE 0,23:PRINT"<L>oad <C>opy <M>ove <D>elete
<R>ename <U>ndelete Drive <0><1><2><3> <I>nstall";
550 FOR X=0 TO 1 STEP 0
560 U$=INKEY$:IF U$="" THEN NEXT X
570 IF U$="U" THEN 7000
580 IF U$>"/" AND U$<"4" THEN D=VAL(U$):DRIVE D:NEXT ML
590 IF FC=0 THEN SOUND 1,5:NEXT X ' ONLY UNDELETE IF NO FILES IN DISK
600 IF U$="L" THEN 4000
610 IF U$="C" OR U$="R" THEN 5000
620 IF U$="M" THEN 5500
630 IF U$="D" THEN 6000
635 IF U$="I" THEN 11000
640 SOUND 1,1
650 NEXT X
1000 ' READ FAT
1010 FC=0:DF=0
1020 FOR X=3 TO 11
1030 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
1040 DSKI$ D,17,X,A$,B$
1050 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0

```

```

1060 C$=A$+LEFT$(B$,127)
1070 FOR A=0 TO 7
1080 FI$=MID$(C$,A*32+1,11)
1090 IF ASC(LEFT$(FI$,1))=255 THEN A=8:X=12:GOTO 1140 ' EXIT LOOPS
1100 FC=FC+1
1110 EX(FC)=ASC(MID$(C$,A*32+12,1))
1120 FI$(FC)=LEFT$(FI$,8)+"."+RIGHT$(FI$,3)
1130 IF ASC(LEFT$(FI$,1))=0 THEN DF=DF+1:DF$(DF)=FI$(FC):FC=FC-1
1140 NEXT A,X
2000 CLS ' LIST FILES
2010 PRINT"DRIVE"D":"FC"FILES,"DF"DELETED"
2020 PRINT"FREE GRANULES:"FREE(D)
2030 IF SW>0 THEN 2080
2040 IF FC=0 THEN PRINT"THERE ARE NO FILES":SOUND 1,4:RETURN
2050 FOR A=1 TO FC:PRINT " ";FI$(A);:NEXT A
2060 FL=2 ' FILES BY LINE
2070 RETURN
2080 PRINT
2090 IF FC=0 THEN PRINT"There are no files":RETURN
2100 IF SW=2 THEN FL=5 ELSE FL=3 'FILES BY LINE
2110 FOR A=1 TO FC :PRINT FI$(A);
2120 IF SW=1 THEN PRINT" "; ELSE PRINT " ";
2130 IF INT(A/FL)=A/FL THEN PRINT
2140 NEXT A
2150 RETURN
3000 ' CHOOSE FILE TO OPEN
3010 TF$="":A=1
3020 FOR X=0 TO 1 STEP 0
3030 IF A<1 THEN A=1 ELSE IF A>FC THEN A=FC
3040 CV=INT((A-1)/FL)
3050 IF SW THEN 3120
3060 POKE CP,143
3070 IF A/2=INT(A/2) THEN CH=1 ELSE CH=0
3080 CP=CV*32+CH*16+1088:IF CP>1535 THEN CP=1535 ' DONT POKE BEYOND
SCREEN
3090 POKE CP,142
3100 PRINT@480,"";
3110 GOTO 3170
3120 LPOKE CP,0
3130 CH=A-1-FL*CV
3140 IF SW=1 THEN CP=442609+CV*80+CH*26 ELSE CP=442849+CV*160+CH*30
3150 LPOKE CP,4
3160 LOCATE 0,23
3170 PRINT BL$;
3180 IF SW=0 THEN PRINT@480,"WHICH FILE? "; ELSE LOCATE 0,23:PRINT"Which
file? ";

```

```

3190 PRINT FI$(A);
3200 FOR LO=0 TO 1 STEP 0
3210 A$=INKEY$:IF A$="" THEN NEXT LO
3220 IF A$=CHR$(13) THEN RETURN ' FILE SELECTED
3230 IF A$=CHR$(94) THEN A=A-FL:NEXT X
3240 IF A$=CHR$(10) THEN A=A+FL:NEXT X
3250 IF A$=CHR$(8) THEN A=A-1:NEXT X
3260 IF A$=CHR$(9) THEN A=A+1:NEXT X
3270 IF A$=CHR$(12) THEN 520 ' CANCEL
3280 TF$=TF$+A$
3290 FOR Z=1 TO FC
3300 IF LEFT$(FI$(Z),LEN(TF$))=TF$ THEN A=Z :GOTO 3030
3310 NEXT Z
3320 TF$="":SOUND 1,4
3330 GOTO 3210
4000 ' LOAD A FILE
4010 GOSUB 3000 ' SELECT FILE
4020 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
4030 ON EX(A)+1 GOSUB 4050,4130,4060,4130
4040 GOTO 4010
4050 LOAD FI$(A),R ' A BASIC PROGRAM IS LOADED, THIS ONE IS DEAD
4060 WIDTH 32:LOADM FI$(A)
4070 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
4080 IF SW=0 THEN PRINT"FILE"FI$(A)"LOADED":PRINT"PRESS ANY KEY TO
CONTINUE":EXEC 44539:NEXT ML
4090 IF SW=1 THEN WIDTH 40 ELSE IF SW=2 THEN WIDTH 80
4100 PRINT"File"FI$(A)" loaded"
4110 PRINT"Press any key to continue":EXEC 44539:NEXT ML
4120 RETURN
4130 IF SW=0 THEN PRINT@496,""; ELSE LOCATE 23,23
4140 PRINT"WRONG FILE TYPE";:SOUND 1,4
4150 TF$="":RETURN
5000 REM COPY/RENAME FILE
5010 NF=FC:FOR A=1 TO 72:WF$(A)=FI$(A):NEXT A
5020 GOSUB 3000 ' SELECT FILES
5030 IF SW=0 THEN PRINT@480,""; ELSE LOCATE 0,23
5040 PRINT BL$;
5050 IF SW=0 THEN PRINT@448,"DESTINATION FILE? "; ELSE LOCATE
0,22:PRINT"Destination file? ";
5060 LINEINPUT DF$:IF LEN(DF$)>14 THEN SOUND 1,4:GOTO 5050
5070 IF LEN(DF$)=0 GOSUB 2000:GOTO 520
5080 'ERROR TRAPPING FOR COCO 1/2, EXPAND FILE NAME TO 8.3
5090 EP=INSTR(1,DF$,"."):IF EP=0 THEN EP=INSTR(1,DF$,"/")
5095 IF EP=0 THEN IF LEN(DF$)>8 THEN SOUND 1,4:GOTO 5050 ELSE
EP=9:DF$=DF$+"."
5100 DF$=LEFT$(DF$,EP-1)+STRING$(9-EP,"")+MID$(DF$,EP,LEN(DF$))

```

```
5110 FOR X=1 TO 72:IF FI$(X)=DF$ GOSUB 8000:GOTO 5050 ELSE NEXT X
5120 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
5130 IF U$<>"R" THEN COPY FI$(A) TO DF$ ELSE RENAME FI$(A) TO
DF$:FI$(A)=DF$:GOSUB 2000 ' UPDATE LIST, NO NEED TO READ THE DISK AGAIN
5140 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
5150 IF U$="M" THEN RETURN
5160 IF U$="R" THEN 520 ELSE NEXT ML
5500 REM MOVE FILE
5510 GOSUB 5010
5520 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
5530 KILL FI$(A)
5540 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
5550 NEXT ML
6000 REM DELETE A FILE
6010 GOSUB 3000 ' SELECT FILE
6020 IF SW=0 THEN PRINT@480,""; ELSE LOCATE 0,23:PRINT"";
6030 PRINT"REALLY DELETE "FI$(A)" (Y/N)?";
6040 FOR LO=0 TO 1 STEP 0:A$=INKEY$
6050 IF A$="N" THEN GOSUB 2000: GOTO 520
6060 IF A$="Y" THEN LO=1
6070 NEXT LO
6080 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
6090 KILL FI$(A)
6100 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
6110 NEXT ML
7000 REM UNDELETE FILES
7010 IF DF=0 THEN SOUND 1,4:GOTO 520
7020 CLS
7030 PRINT"DELETED FILES IN DRIVE"D
7040 IF SW>0 THEN 7070
7050 FOR A=1 TO DF:PRINT DF$(A),:NEXT A
7060 PRINT:PRINT"WARNING! UNDELETE IS A 'BETA', AND WORKS ONLY WITH 1
GRANULE FILES"
7070 FOR A=1 TO DF:PRINT DF$(A);
7080 IF SW=1 THEN PRINT" "; ELSE PRINT" ";
7090 IF INT(A/FL)=A/FL THEN PRINT
7100 NEXT A
7110 PRINT:ATTR 0,0,B:PRINT"WARNING":ATTR 0,0:PRINT"Undelete is a 'beta'
and works only with 1 granule files"
7120 GOSUB 3000 ' SELECT FILES
7130 LOCATE 0,23
7140 PRINT"FIRST LETTER FOR FILE?"
7150 FOR LO=0 TO 1 STEP 0:SN$=INKEY$:IF SN$<>" " THEN LO=1
7160 NEXT LO
7170 FOR X=3 TO 11
7180 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
```

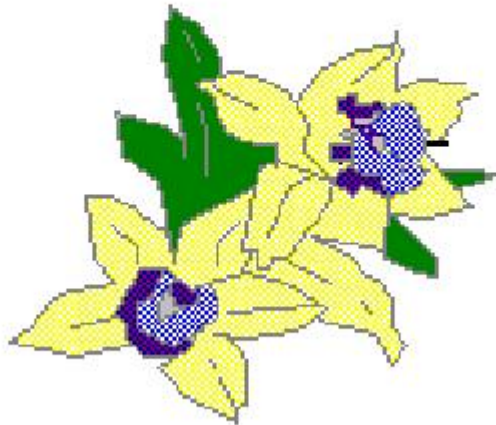


```

7190 DSKI$ D,17,X,A$,B$
7200 C$=A$+LEFT$(B$,127)
7210 PF=INSTR(1,C$,LEFT$(DF$(A),8))
7220 IF PF=0 THEN NEXT X
7230 MID$(C$,PF,1)=SNS
7240 A$=LEFT$(C$,128):B$=RIGHT$(C$,127)
7250 DSKO$ D,17,X,A$,B$
7260 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
7270 NEXT ML
8000 IF SW=0 THEN PRINT@448,"FILE ALREADY EXISTS <O>VERWRITE ?"; ELSE
LOCATE 0,22:PRINT"File already exists, <O>verwrite?"
8010 SOUND 1,4
8020 FOR X=0 TO 1 STEP 0
8030 A$=INKEY$:IF A$="" THEN NEXT X
8040 IF A$<>"O" THEN RETURN
8050 KILL DF$:GOTO 5120
9000 REM ERROR TRAPPING ON COCO3
9010 EN=ERNO:EL=ERLIN
9020 IF EN<>20 THEN 9050
9030 PRINT"INPUT/OUTPUT ERROR WHILE WORKING WITH THE FILE "FI$(A)
9040 PRINT"YOUR DISKETTE MIGHT BE FAILING,CONSIDER REPLACING
IT":GOTO 9080
9050 IF EN=33 THEN 8000
9060 PRINT"UNEXPECTED ERROR #"EN" IN LINE"EL
9070 PRINT"THIS MIGHT BE A 'FILE NOT FOUND' ERROR"
9080 PRINT:PRINT"PRESS ANY KEY TO TRY TO CONTINUE"
9090 SOUND 1,4
9100 EXEC 44539
9110 GOTO 490
10000 CLS:PRINT"Thanks for using X-Filer v.2.3"
10010 PRINT"Please remember that the program is still loaded in memory and that your CoCo
isoperating at double speed"
10020 PRINT "Free memory:";MEM
10030 CLEAR 200:END
11000 CLS
11010 IF FREE(D)<3 THEN 11070
11020 IF SW=0 THEN PRINT"INSTALLING X-FILE":PRINT"TO DRIVE";D ELSE
PRINT"Installing X-FILE to Drive"D
11030 IF C3 THEN POKE 65496,0 ELSE POKE 65494,0
11040 SAVE"XF"
11050 IF C3 THEN POKE 65497,0 ELSE POKE 65495,0
11060 NEXT ML
11070 IF SW=0 THEN PRINT"NOT ENOUGH SPACE IN THIS DISK TO INSTALL
X-FILE" ELSE PRINT"Not enough space in this disk to install X-FILE"
11080 SOUND 1,5:EXEC 44539
11090 NEXT ML

```

Well here it is another one. This one did not make it as a Christmas or even a New Years issue. I guess I could have decorated it with a Valentines Theme huh. I am sitting here now wondering if maybe this project has lost momentum or not. This E-Zine would have never grown to what it is today without everyone pitching in. The only thing I should be credited with is just pulling together all the programs and articles into one readable format. Although trying to get Microsoft (sorry for cussing ☺) Word to do what I want it to is sometimes a huge challenge in itself. My original focus for doing this project has evolved over time. At first I just wanted to relive the closeness of the people like in CFDM, but now I want to leave something for my son to see if I am one day not around to be with him anymore. I lost my dad when I was young, and I do not have much to go on. I wonder sometimes what was he like, did he chew his fingernails like my brothers do, was he a good swimmer like me, was he forgetful like I am, is he a procrastinator or is he a person that goes and gets the job done ASAP. I look at my son and hope that one day he will not be searching for those answers about me. I hope that if I am not around he will be able to see through my work a little piece of my life that my family knows very little about. I have recently learned that my dad was a big time environmentalist, and that he loved to surf. Looking at the picture I have of my dad, he sure does not look like a surfer. Well anyway I hope that one day your kids can look through the internet and when they google your name they come across these newsletter or E-Zines and see some of the things you have wrote. This is not a big shot magazine, but unlike a magazine anyone can google and find you here.



Drawing produced on a Coco3 by Graham Elphick from Australia.