CoCo~123
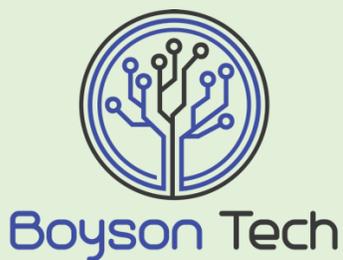
A publication of the Glenside Color Computer Club

Spring 2021 edition

Products, Services, and Content

Boyson Tech

https://boysontech.com/marketplace

Cloud-9

http://cloud9tech.com/

COCOMAN.BIZ

maker of the switch-a-roo

http://cocoman.biz

COCOVGA

http://cocovga.com/

RETRO INNOVATIONS

http://store.go4retro.com/

The Zippster Zone

https://thezippsterzone.com/

The CoCo Crew Podcast

J. Linville   N. Blanchard   M. Rowen

www.cococrew.org

http://cococrew.org/

COCO TALK LIVE

http://cocotalk.live/

WynDee Systems

TRS-80

RETRO COMPUTER SOLUTIONS

http://www.tindie.com/stores/fiscap0768/

NitrOS-9

Ease of Use Edition

http://lcurtisboyle.com/nitros9/nitros9.html

Memory Map

G.C.C.C. Meetings

GCCC meetings are the 3rd Thursday of each month and are being held virtually via Blue Jeans until further notice. Upcoming meetings: April 15th, May 20th, June 17th, 2021

G.C.C.C. OFFICERS

If you have questions about the association, contact one of the officers for the answers.

POSITION	NAME	E-MAIL	PRIMARY FUNCTION
President	Jim Brain	president@glensideccc.com	The buck stops here...
Vice-President	Terry Steege	vice-president@glensideccc.com	Meeting planning, etc.
Treasurer	Brian Goers	treasurer@glensideccc.com	Dues and Purchasing
Secretary	Rich Bair	secretary@glensideccc.com	Records and Reporting

Jim Brain   Terry Steege   Brian Goers   Rich Bair

From the President's platen

If I'm honest, I both love and loathe when Spring rolls around. Obviously, after being somewhat cooped up inside during the Winter (even before current events pushed us all indoors), it's so nice to enjoy some fresh air and the ability to step outside without worrying about frostbite or the heft of winter coats, scarves, gloves, and hats. On the other hand, it also means the need to split time between computing projects and outdoor efforts (I have not yet figured out how to enjoy my computing hobby while in the backyard, though others may have solved that puzzle). Winter is my most productive classic computing time, enjoying these machines and the relationships that accompany them. There's still the evenings, though, when the Sun disappears below the horizon. I can then continue my fascination with these machines and those who use them, author software on them, manufacture peripherals for them, write about them, demonstrate features on them, and present about them.

I know our newsletter staff are preparing another enormous newsletter feast for you, and I'm sure more details on these items appear elsewhere, but let me at least touch on some ongoing work:

- Though we've made the difficult decision to postpone the 2021 CoCoFEST, we are still committed to holding a face to face event in 2021. If you have not already, please update your calendars to join us at the Elk Grove Holiday Inn on November 6 and 7 (with setup on the evening of the 5th). It's a new venue, a new year, and hopefully a new view into the future of CoCo!
- At the inception of the group, it was always intended we would be a non-profit 501 c(3) organization. Donations would be qualified for tax deductions, the organization could take advantage of services for non-profits, etc. For whatever reason, that step was never completed, and there's no better time than the present to address this. As you can expect, it's more complicated than just checking a box on a form, but we're working though the process.
- Though it predates our 501 c(3) journey, non-profit requirements will mandate some adjustments to our constitution and bylaws. We'd already started the process to update them, so we're continuing that process.
- To both foster some community among our now geographically varied membership and offer an opportunity to bring new software or hardware to the attention of the club, we've started a demonstration time 1 hour prior to the monthly business meeting (6:30PM Central). As with the business meeting, attendance remains optional, though early results show promise. Over time, I hope to attract content creators to demonstrate their software and/or hardware during this portion of the monthly meeting.
- As some of you might recall, Glenside's December 2020 virtual social gathering was enjoyed by many. While the monthly demonstration time also supports a social aspect, the postponement of the CoCoFEST from April opens up an opportunity for a updated virtual social gathering. If you're not otherwise busy, I hope you can join us for a moment or two. Watch the website for the specific date and time.

If you're interested in helping or hearing more about any of the current efforts, feel free to drop me a note via the web site or via email.

Though I'm not the best Marketing resource in the community, I'm trying to keep our various social spaces (web site, Facebook, mailing lists, etc.) updated with current information. I hope you check one of those resources periodically to stay up to date on the club's current events and efforts.

Jim Brain
Glenside President 2020-2021
president@glensideccc.com

Secretary's Scroll

The principal topic in each of the last three membership meetings has been "What about the CoCoFEST!?"

By the time this column is published there may be an answer, but at this moment I don't know. At our February meeting we started with 8 options, which we quickly winnowed down to 5 by eliminating "leave it in April", "reschedule to May", and "reschedule to late November". The five remaining options were put on TandyList in a format where contributing members could vote their preferences. Voting is scheduled to close by the beginning of March because the Holiday Inn Elk Grove Village demands a reply by that time, so I hope that all of you will have made your opinions known before then.

Despite the distraction of the pandemic, Eric Canales, our Director of Digital Media, has finished the project of getting our web site fully under club management and control, and he has completely redesigned it as well. Forward your comments and ideas to him through the link in the "About Us" directory of our web site (www.glensideccc.com). Randy Weaver, our Director of Information Services, continues to make improvements to his TandyList with the goal of eventually having it fully integrated with the official club web site. He would also appreciate your opinions on his work. And finally, Steve Strowbridge, our newsletter editor, is extremely proud of this issue in which you are reading my column.

Contact information for Randy and Steve is also in "About Us" on the web site.

At the January meeting it was suggested that we initiate a custom of having the official monthly meeting (7:30 P.M. CST) preceded by an hour or so of unstructured meeting time when attendees could converse on any CoCo topic of their choice, present demos, etc. This took place before the February meeting, and participants seemed to enjoy it greatly. So next month, consider tuning in at 6:30 PM (CST) and see what ideas are in play prior to the meeting.

Happy CoCoing!

Rich Bair, Secretary
Glenside Color Computer Club
secretary@glensideccc.com

Webmaster Update

Hello all! The new website (https://www.glensideccc.com) has been up and running for a few months now and we've added a lot of stuff, so make sure you take a look.

One thing to make sure to check out is under Documents > Newsletters. We have newsletters uploaded dating back to 1985!

We have a few missing editions, so if you have any copies of the missing newsletters, make sure you forward them to me.

We are keeping the CoCoFEST! 2021 information up to date, and we have news posted as things are happening so keep an eye out!

Eric Canales
GCCC Webmaster
webmaster@glensideccc.com

R.I.P. Michael Knudsen

On March 9, 2021, Michael Jeremy Knudsen, PhD, age 77, past president of Glenside Color Computer Club, CoCoFEST! auctioneer and writer of the CoCo/OS9 based music composition program, ULTIMUSE, passed away.

Beloved husband of 50 years to The Rt. Rev. Chilton R. Knudsen; Loving father of Daniel Jeremy Knudsen of Portland, ME; Dear brother of James Knudsen and wife Raina of Aiken, SC and brother-in-law of Ward Richardson and wife Sharon of Philadelphia, PA, James Richardson and wife Leslie of Atlanta, GA, Laurie Richardson and husband Friedl Pelosheck of Vienna, Austria;

Mike is also survived by nieces, nephews, and many wonderful friends across the US; Mike was preceded in death by his parents, Mildred (nee Benson) and Harvey Peter Knudsen, and by his brother, Peter Knudsen.

Due to capacity limits, the Funeral Service is by invitation only on Saturday, March 20, 2021, 11:00AM, St. Bartholomew's Episcopal Church, 4711 Edmondson Avenue, Baltimore, MD 21229. If you would like to join the service, click on "Join Livestream" on the funeral home website or to view a recording of the funeral service, please visit the church website, https://stbs-md.org/

Mike will be laid to rest at the Cathedral Church of St. Luke in Portland, ME.

In lieu of flowers memorial contributions may be made to the St. Bartholomew's Episcopal Church or to the Parkinson's Foundation, https://www.parkinson.org/ways-to-give.

To Plant Memorial Trees in memory, please visit our Sympathy Store.

Published in Baltimore Sun on Mar. 17, 2021.
Submitted by Tony Podraza

Feedback!

Wow. That's the closest I've gotten to reading Rainbow in decades. So much good content. The yearly news recap is outstanding.

-Allen Huffman

Very impressive work! Kudos to everyone involved in the making of this publication.

-Peter Cetinski

Outstanding newsletter. If this keeps up you'll need more staff. CoCo Ho!

-Rick Ulland

Steve, Hope you don't get burned out doing this. As Allen stated this is the closest thing to the Rainbow ever. I do have a couple RSDOS program I've altered to suit me that I would be willing to contribute after I dig them up. Merry Christmas

-Richard Crislip

Yeah! I saw your article in there, Rick [Adams]. Great to see your name "in print" again. I haven't been published in a CoCo thing in many years, so I'm thinking of submitting some things for consideration as well.

-Allen Huffman

I just sent the file off to get printed and bound so I can have a physical copy for myself. That was a nice issue. If we get a year long retrospective every year they will be nice keepsakes

-Michael Brant

Haven't read it in any detail yet but after reading Allen's comment I had to see for myself what all the fuss was about... WOW!!! This really is reminiscent of the old CoCo magazines! Outstanding effort all involved!

-Mark McDougall

Awesome newsletter

-Mark Adams

Just the BASICs: A couple of Graphics Concepts

by Alison deNu

There are a few concepts that are extremely useful to programming a graphical arcade-type game, such as measuring distances and using angular movement. Let's skim these two concepts, shall we?

Let's say you're programming a game where you need to know if the Player character's avatar is within "agro distance" of an enemy unit. How do you quickly and easily determine the distance between two objects on a two-dimensional grid? That's easy, thanks to that old dead Greek dude, Pythagorus!

That grid is just your old familiar two-dimensional Number Line from school arithmetic. And we've made ours even simpler because we eliminated all of the negative numbers and only deal with the positive quadrant:
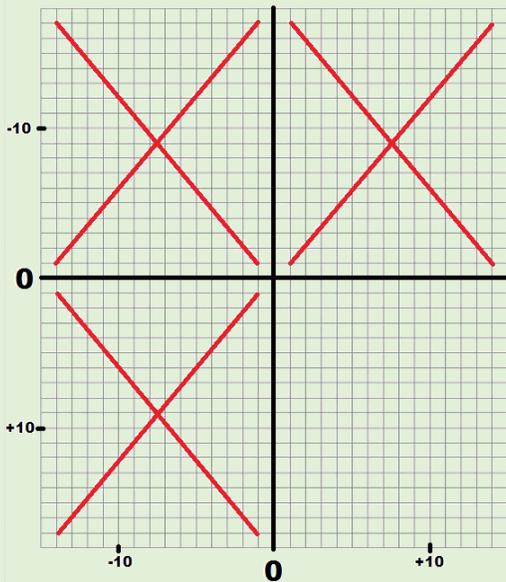
So our grid looks more like this:
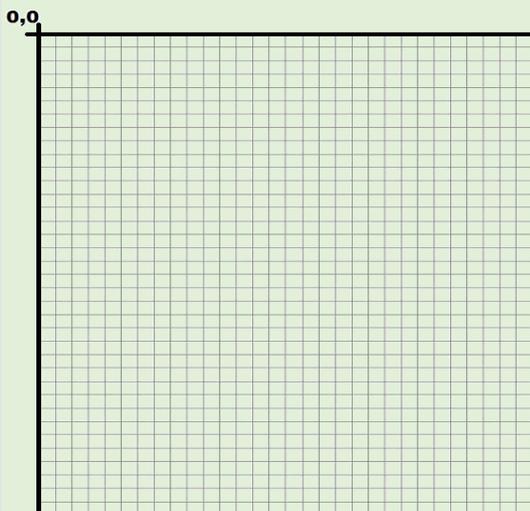
Now, let's imagine two objects on this grid, Red Object and Blue Object:
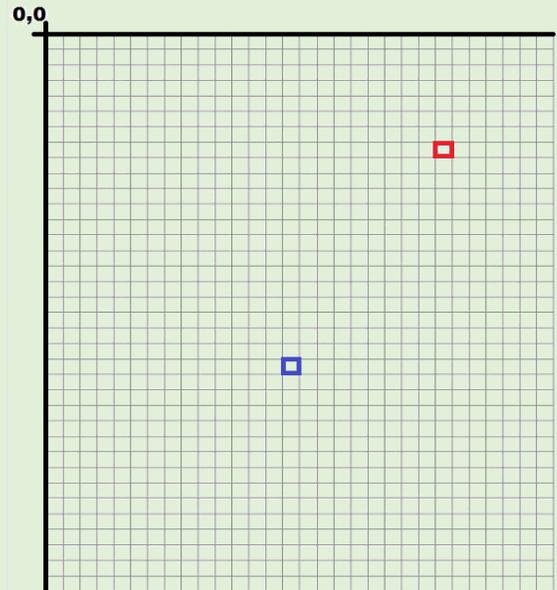
The Red Object is at 24,8 and the Blue Object is at 15,22. Now, how do we determine the exact distance between these two objects? We use the old Pythagorean Theorem, of course! To do that, we find the place where the X axis of one object intersects the Y axis of the other object. That will give us our third vertex, the first two being the locations of the two Objects:
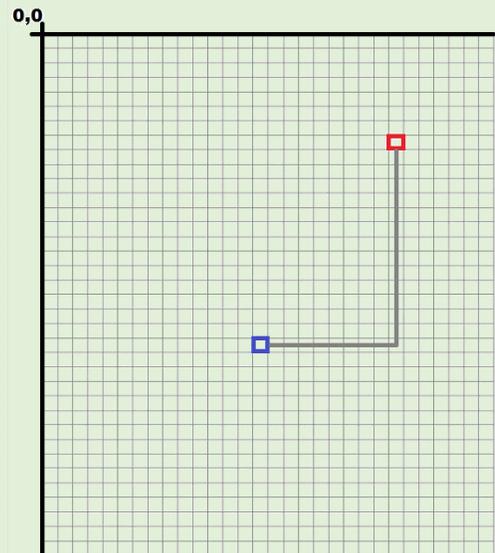
In this example, the vertex where the X axis of the Blue Object intersects the Y axis of the Red Object gives us 24,22. We could have instead chosen the vertex where the X axis of the Red Object and the Y axis of the blue object intersect, at 15,8, to derive the same ultimate answer.

Your choice of which object provides the X or Y axis is completely up to you, the answer will be the same regardless:

0,0
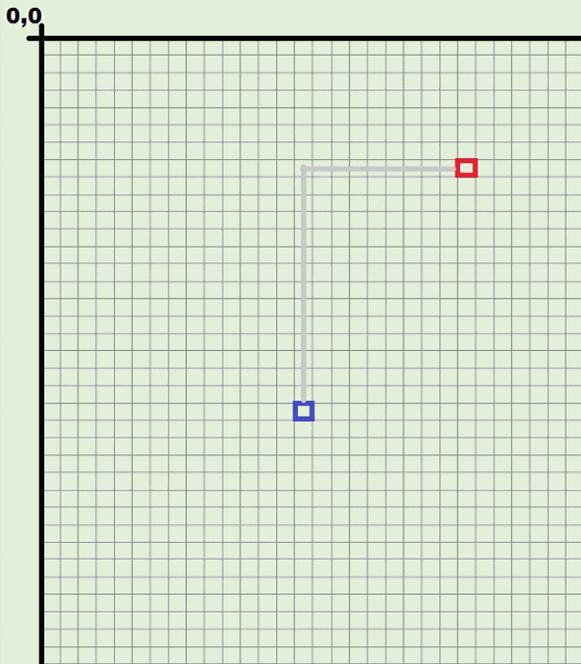
In our example where the third vertex is at 24,22 the X distance from the third vertex is 9, and the Y distance from the third vertex is 14. Pythagorus tells us that if we square those distances, and add them, we will have the square of the distance between the Red Object and the Blue Object. So 9x9=81 and 14x14=196, therefore the distance between the Red Object and the Blue Object is the Square Root of 81+196, which is √277, which is 16.64. So the Red Object is about 17 pixels away from the Blue Object according to that dead Greek dude:

0,0
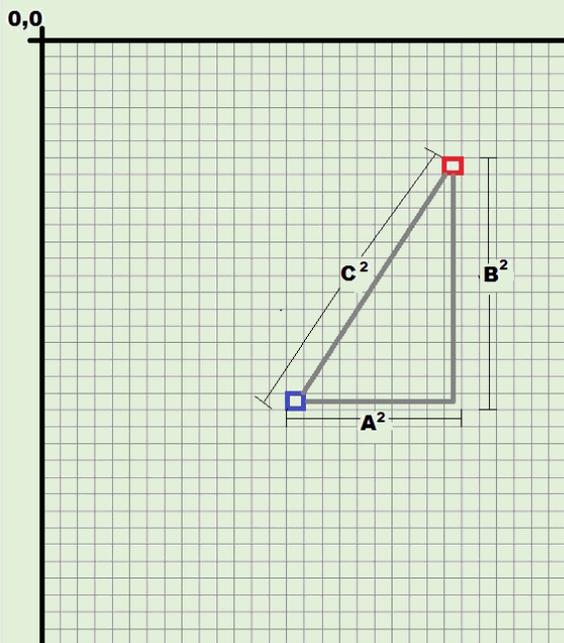
Now, about that angular movement thing. How do we generate an angle in a two-dimensional grid space? Turns out that it's easy because the computer does all the math for you. Take the Sine and Cosine of an angle, let's say for example 90 degrees. The Sine of 90 degrees is 1, the Cosine is zero. Sine returns a number between zero and one, and so does Cosine. Sine returns a value related to the X axis, and Cosine returns a value related to the Y axis. **That number represents the position of a point in 2D within a single pixel's space:**
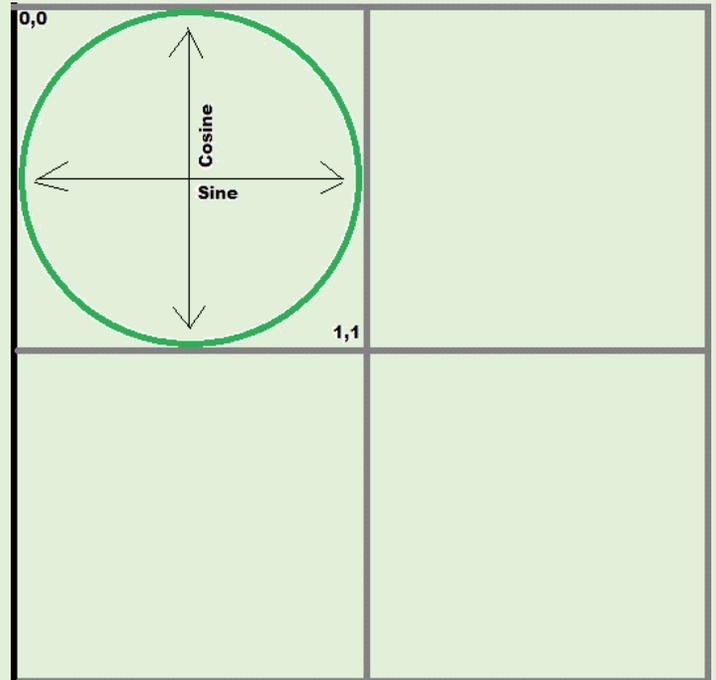
What this means is that the Sine and Cosine of any angle given will result in an X and Y position **within a single pixel** for that angle:

If the angle given is 45 degrees, we get a Sine of about .75, and a Cosine of about .25

If the angle given is 270 degrees, we get a Sine of 0, and a Cosine of about .5
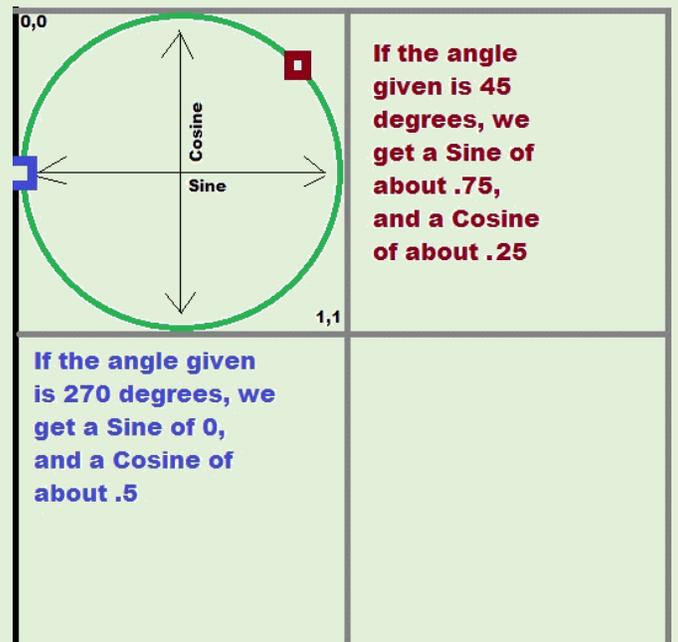
Now let's say we want to adjust both the position and scale of that angle. That's easy, we multiply times the radius in pixels, and add the desired X and Y origin positions, in that order.

For example, let's say we want to draw a circle:

```
10  PMODE 4,1
20  SCREEN 1,1
30  PCLS
40  FOR A=0 TO 360
50  X=SIN((A/57.29577951))
60  Y=COS((A/57.29577951))
70  XP=17
80  YP=14
90  R=10
100  PSET((X*R)+XP,(Y*R)+YP)
110  NEXT A
120  GOTO 110
```

This example program should draw something like this:
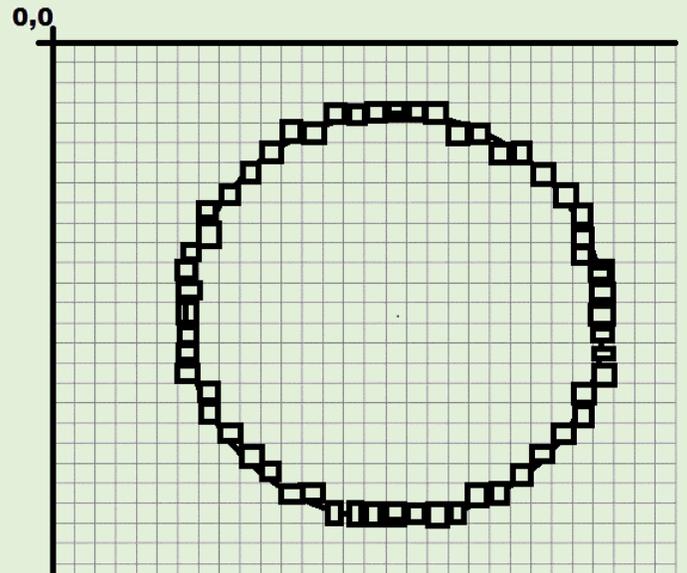
And now you know two of the simpler ways to program using the math of two-dimensional grid type graphics.

I hope that this helps you to code some awesome Coco games for us in the near future!

CoCo 1 Adventure

Part 1: Initial problems

Salvador Garcia

Introduction

I was in Mexico when the Glenside Color Computer Club held its first ever Holiday Virtual Open House on December 20, 2020. That was a good thing because that is where I have my CoCo 1 stored. I decided to unpack it and get it back in working order.

Along with the CoCo 1, I also have a 26-3022 floppy drive subsystem which I haven't used since the mid-1980s.

I used an RCA to RF adapter to connect the CoCo to the TV's RF input.

This method Just Works, but not very well. The screen, while legible, is littered with interference. I was not concerned about it at this time as my main concern was to get the CoCo operational. Once I have everything in order, I'll think about modding the CoCo so that its screen displays in higher quality. As I set up the system and powered it on, I found various issues. This first part chronicles these issues and includes the feedback I received when I asked on the Facebook group and CoCo List.

The CoCo 1 flap

The first issue that I noticed was the CoCo's cartridge flap. The flap is hinged, and a spring keeps it closed. When a cartridge is inserted, the flap is pushed back and up to the open position and the spring is tensed. Once the cartridge is removed, the spring pushes the flap back down to the closed position.

The problem with my CoCo is that the flap did not come back down when the cartridge was removed. In fact, the flap would stay at the position that I placed it moving it manually. It seemed like the spring had lost its tension.

In the above image, the cart's connector is easily seen because the flap was pushed open and it stayed in this position, even without a cartridge.
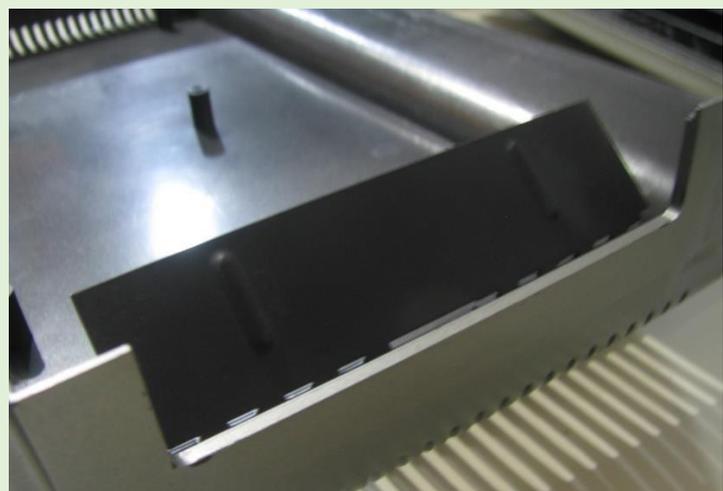
In the above image, I position the flap half-way and it stayed there! I asked about this on the CoCo List.

Arthur Flexser suggested that the spring might have popped off and recommended opening the computer and locating the spring. He also suggested that the problem might be a dirty hinge. I removed the CoCo's cover to have a better look. I checked the spring and to my surprise I found that the spring still had its tension and was in its proper place. This was not the problem. In fact, the problem was a dirty hinge that caused the flap to stick.

Arthur further suggested applying some light lubricating oil. Frankly, I didn't have the heart to "oil" the hinge. I've never been partial to oiling hinges that were not originally oiled. I then got an email from Joe Schutts. He recommended cleaning the hinge with denatured alcohol and a toothbrush. Being on vacation, I wasn't sure I had the alcohol, but was lucky enough to have some. No toothbrush though, so I used a Q-tip.

To my surprise, the flap moved freely when I cleaned it with the alcohol but stuck again once the alcohol dried. Okaaay. Not having many cleaning options, I decided to defer this maintenance till later. I thought of removing the hinge wire to better clean it, but I did not see any obvious way of doing so. I decided to leave it as is for now. I didn't want to risk making the issue worse.

Keyboard Funkiness

Another issue I noticed was with the keyboard. Some of the keys did not travel freely and seemed to get stuck halfway, not traveling all the way to register the keystroke. Some keys that did travel freely did not register the keystroke anyways.

I tried entering a short BASIC program, but I would miss many keystrokes. I wasn't able to find a pattern to the missed keystrokes because this issue was intermittent. The keys that did not register the keystroke would work fine later.

I noticed that this problem diminished when the 20-3022 floppy controller cartridge was removed. I had also asked about this on the CoCo list and Arthur suggested that I reseat the chips both on the CoCo and the 20-3022 controller. I did so.

Arthur also suggested swapping out the 6809; however, being away from home, I did not have another 6809 lying about. Swapping or replacing the CPU was not feasible at that time, but I made a mental note to pack a 6309 the next time that I

travelled to Mexico. Fortunately, the CoCo 1's processor is socketed, so it is just a matter of swapping out the chips.

The keyboard issue seemed to diminish as I used the CoCo. I was not sure if reseating the chips helped or whether it was another, as of yet unidentified, issue. I decided to locate a spare keyboard so that I could swap them out and see whether this made a difference. Henry Rietveld came to the rescue, offering me a free CoCo 1 keyboard for just the shipping costs from Canada. I jumped at the chance! Once I have the keyboard, I'll be sure to bring it along the next time that I travel to Mexico.

I checked the keyboard connector. To my surprise, it used a header connector, not the thin film connector that I've seen on later model CoCos.

The image above shows the keyboard male header that's on the CoCo's motherboard. The female header is on the keyboard, using a short stiff flat cable that goes into the keyboard, as shown in the next image. Swapping the keyboard will be easy enough and it will give me a good idea where the problem is located.
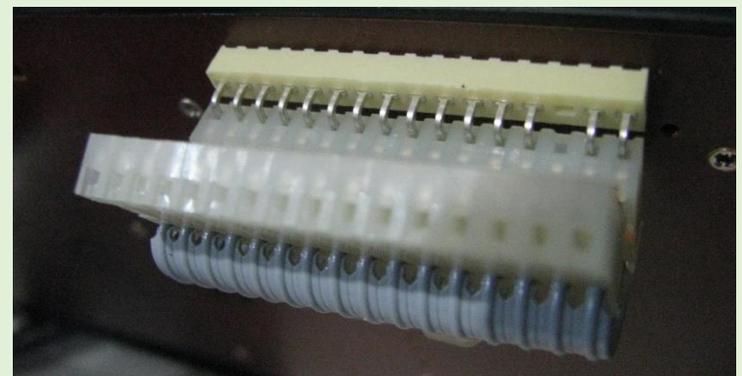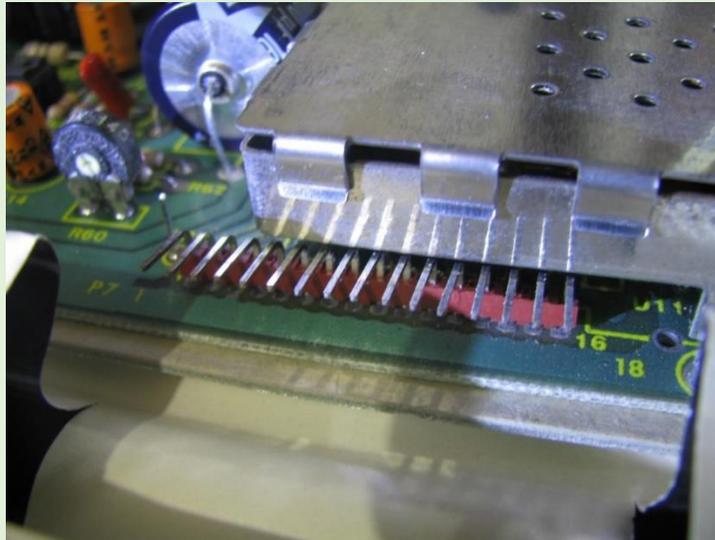
Floppy drive Misbehavin'

The next step was to format and perform some I/O on a floppy disk. This was easier said than done! The first hurdle was locating a floppy diskette! I rummaged through some boxes until I finally hit the jackpot! A box was mostly filled with diskettes. Many of them were used, but I found some that looked brand new.

I took one and inserted it into the drive. I used the disk subsystem's reference guide to determine how to insert the floppy. I found an illustration that showed a diskette being inserted, with the read-protect notch at the top, so that is the

way I inserted the diskette. I anxiously went to the keyboard and entered the following command:

DSKINI

The interference on the screen increased for a few seconds, but I did not see any movement on the drive, except that the motor began to spin-up. The drive's LED did not light. Then after a few seconds the ubiquitous message displayed:

IO ERROR

I reexamined my connections and made a list of possible problems:

1. Dirty card edge connectors on the cartridge and drive.
2. Bad cable or cable connected backwards
3. Bad floppy drive
4. Bad cartridge
5. Bad floppy diskette

I started out by cleaning the card edge connectors on the drive and cartridge.
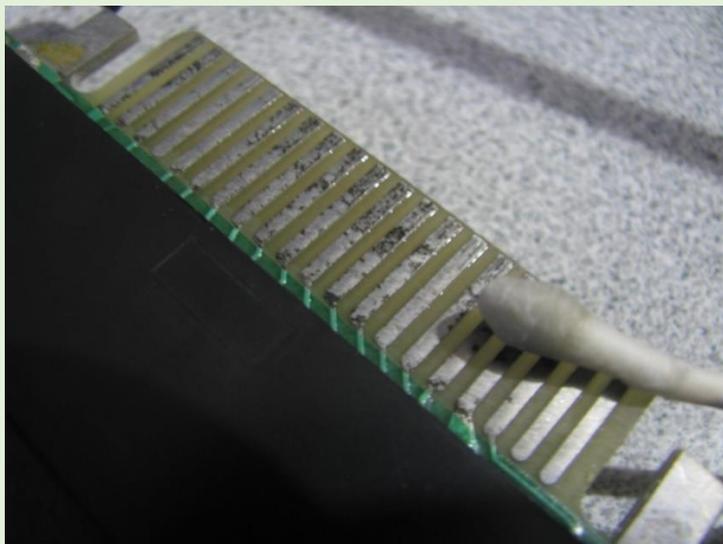
Some of the connectors were dirty. The Q-tip became soiled as I gently stroked the metal strips. Note: Q-tips aren't the best tool for cleaning. They bend easily, even with light pressure. After cleaning them I retested, making sure all the connectors were dry. This had no effect on the drive's behavior.

Next on the list was the cable. I had possibly connected the cable backwards. I consulted the disk subsystem's reference guide, but I did not see any mention or illustrations showing the orientation of the cable, just that the drive should be connected to the innermost connector (B) from the drive end of the cable. I assumed this was for drive 0 and the connector at the end of the cable was for drive 1. The image above shows the drive cable, labeled using the same nomenclature as used in the Tandy reference manual.
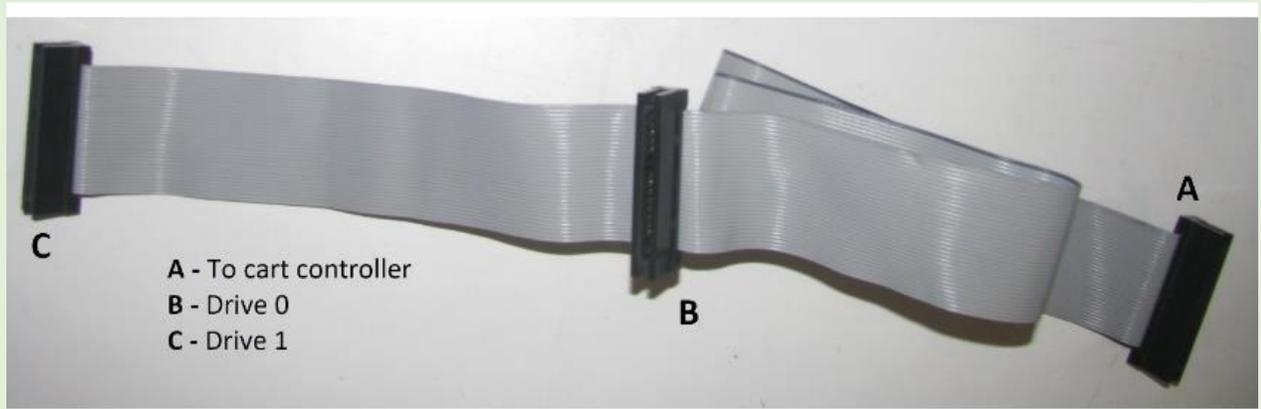
Looking at the cartridge, I couldn't find any indications to tell me which was pin 1. Since I had opened it to reseat the integrated circuits, I quickly removed the cover and had a closer look. I located pin 1 for the flat cable. It was located at the rear of the controller cartridge once it was plugged in.

I have found that pin 1 is usually oriented towards the top of the drive, so I assumed that this was pin 1. When I had a closer look at the floppy drive, I saw the notch on the card edge connector, but it was at the bottom. I couldn't see the silk screen, so I removed the drive's cover and on closer examination I realized that pin one was at the bottom of the drive! I had the cable connected backwards.
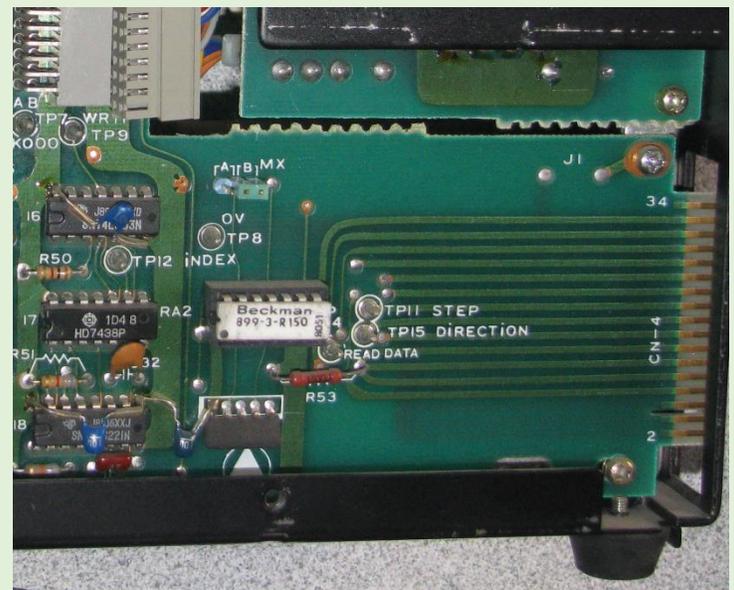
The above image clearly shows the floppy drive's edge connector and pin 2 at the bottom of the drive and close to the notch. Pin 1 located!

I then correctly connected the cable and retested. This did not improve the behavior of the DSKINI command. The drive was

now being addressed correctly, and the screen no longer displayed the interference it did before, but the drive made no attempt to format the diskette.

Knowing that I had performed the basic steps, I decided to ask on the CoCo List and on Facebook. The replies poured in! Thanks again everyone! The fist reply came from Bill Owens:

"Speed adjustment under a florescent light is easy to check"

Sigh. I did not have a florescent lamp! OK, no problem! For now, I'll have to assume that speed is not the problem. When I have exhausted all options, I'll figure out where I can get a florescent lamp.

Robert Schofield chimed in with a wealth of information:

"I assume you are doing the usual - turn on the disk drive unit first, then CoCo?

"Powering on a drive unit with a good power supply should cause the select light to come on briefly and spin the motor, even if the computer is not powered on.

"If the motor does not spin the disk and the select LED does not come on, I would investigate the power supply/fuse.

"From your description, the MOTOR SEL line between the CoCo and the disk pack seems to be connected, and the PSU seems to be working.

"However, it's not clear if the disk drive spins up when you try to access the drive: can you clarify? If it does not spin up then you have a motor fault in the drive, which could either be a broken belt (non-direct drive) or a sticky belt (preventing the motor spinning) or a failed motor (direct drive types).

"To me, a DIR listing of the drive is the simplest READ access of the disk, so it should SELECT and spin up for a few seconds, with a click as the heads pull in and the usual whispering sound you hear from the heads as they track over the disk surface. Do you get any of this?

"If you do, and you get an IO ERROR, then either the disk is unformatted or there is potentially a head defect, most likely iron oxide on the head face, or an alignment problem, or a sticking head carriage."

Thankfully, I was powering on the system correctly: Drive first, CoCo second. I didn't have a multimeter, so measuring voltages was out of the question, but the drive did power on correctly, so I assumed that the power supply was good.

Meanwhile, on the CoCo List Arthur Flexser mentioned that the problem might lie with the HALT line of the 6809 and suggested that I swap out the 6809. L. Curtis Boyle backed up this recommendation. I was woefully unprepared to swap out components. But as I mentioned earlier, I will bring spares with me next time I am here.

Both Robert Schofield and Steve Norton asked about the cable:

"I didn't think to ask: how many drive connectors are on your cable from the controller, one or two? Into which connector is the drive plugged?" - Robert Schofield

"How many connections on your disk cable? Any chance that the cable is backwards?" - Steve Norton

This was an interesting question. My cable, as noted, had two connectors and I had my drive connected to the innermost connector. I had thought about moving the drive to the outermost connector, but I wasn't sure if I needed to have a drive on the innermost connector before I did this. I asked to confirm. Steve got back to me indicating that I should connect the drive to the outermost connector.

I retested, but it didn't seem to make a difference. DSKINI and DIR had no effect. More replies poured in:

"Sometimes there is a termination resistor pack on the floppy drive. Only the last drive on the cable has the pack, so if the drive was ever used in a system with more than one drive its resistor pack might have been removed, and that could cause some issues. If the light does not come on that could also be a missing jumper for "drive select" on the floppy drive. But neither of these should be a problem if the drive was only ever used solo." – Chris Burke

My CoCo setup was in its original configuration. I never had more than one drive. I was sure that this was not a configuration issue.

"If the drive spins but the disk LED does not light it may be that the drive select is not properly set. A single drive should have its jumpers set for DS0 and the ribbon cable should not have a twist in it." – Barry Nelson

I didn't think it was a configuration issue but decided to look anyway. I did not find a DS0 jumper, but I did find an A/B MX jumper with the jumper's position selecting "B". Not sure what this jumper was, but I decided to change the jumper location to "A". Not good. Now the drive got selected independently of whether I addressed drive 0 or drive 1. Its behavior got weird, so I decided to return the jumper to its original position and leave it like that.

"Please open up the floppy drive and have a look at the logic board. TEC drives used with 26-3022s have a half dozen small capacitors on the logic board and they are notorious for leaking and destroying the logic board. Double-sided TEC-503s are used in Model 4Ds and they ALL die. I am not a "recap everything" person but with TEC drives it's a matter of when not if." – John Wiggins

Interesting. I had given the floppy drive electronics a cursory look and did not spot anything unusual, but I had a second look, this time focusing on any possible capacitor decay. Nope! All caps looked good. No leaks, and no inflated caps.

I retested. This time it occurred to me to enter DIR 1. This made a substantial difference! The drive LED came on and the drive attempted to do something. I then tried DSKINI 1. The drive started to format the diskette. I even counted the number of times the head moved: 34. Yes! Thirty-five tracks and all of them were sought out correctly. But my songs of victory were premature. Once the formatting phase finished the heads were moved back, then after a few seconds I got the IO ERROR again. I reported this on the Facebook group.

"Interesting, a single drive should be 0, not 1, it's also odd that it would matter which connector you use on a flat cable, unless it is a cable with pins missing...?" – Barry Nelson

"If it's a two-position cable, then the drives probably won't be jumpered for select, it will be done with cable select.

"Cable select was very much an IBM-PC style to slightly reduce their production costs (one standard drive configuration) with many other home computer systems using jumper select with a flat cable. I must admit I am struggling to remember the details now - I used to know this stuff off by heart<sad face>" – Robert Schofield

Yup, some of the cable's pins were missing.

Barry then provided an interesting note:

"On some TRS-80 systems, the drives were setup as all drive ids selected, and the cable has the drive select pins removed for all but one drive number in each connector. If your drive is jumped to respond only as drive 1 and your cable is also setup to restrict the drive by connector; that could be why the drive only works in one connector." – Barry Nelson

I began to suspect the cable and made it known on the Facebook group. Barry again chimed in:

"Salvador Garcia Quote: I am suspecting that perhaps it is the cable as the drive should have been recognized as drive 0.

"That is a strong possibility." – Barry Nelson

My next step was to get a replacement cable to test. I had planned in contacting the Color Computer community to see if anyone had one to sell; however, Barry once again chimed in. I could use a normal flat cable with one connector at each end. I worried about removing the connector's pins, but Barry quelled my concern:

"Don't worry about that.

"1) You only have 1 drive so it won't matter.
"2) You could change the jumpers on the drive or put tape over the pins if you ever get a second drive.
"3) If you insist, needle nose pliers and elbow grease?<smiley face>" – Barry Nelson

Mark Bosley also confirmed what Barry had stated:

"You can use that drive with a cable that has all pins. Side effect is you could only use the 1 drive and it would respond as all drives, 0-3." – Mark Bosley

No problem for me. I only had one drive and did not have any intention of getting another one.

Barry also provided links to ebay auctions for the 34-pin connector and some flat cable. Unfortunately, someone beat me to the connectors, but I found a 34-pin flat cable with the right connectors, so I ordered that. This cable does not have the twist and all the connector's pins are intact, but this will not be a problem.

"You may have more than 1 problem here. I would start with the activity light problem. look at the drive cable. it should be set up with a connector on 1 end, then 2 or more connectors closer to the other end. the lone connector goes to the controller. look at the other connectors. are there pins missing?" – Mark Bosley

I agreed. I too suspected multiple problems. I just needed to ask myself, 'What is the next step?' I had already determined that the cable was on backwards and corrected this issue. Connecting the drive to the outer most connector solved another problem, but left a question: Why didn't the drive work as drive 0 on the innermost connector?

Mark was correct. The connectors on the cable were missing pins. Mark posted another comment:

"If there are pins missing you will see that that connector has different pins missing. If so, the connector closest to the controller is 0 then the next is 1 and so on. If all connectors have all pins then the jumpers need to be set in the drive for the drive select. Stock Tandy drives were meant to be selected by the missing pins in the cable. Also, the last drive on the cable needs to have a terminator resistor installed in it. A

terminator can look different for different drives. It would be in socket near the connector." – Mark Bosley

My drive did have a terminator. It looked like a white integrated circuit and was labeled "Beckman", most likely its manufacturer.
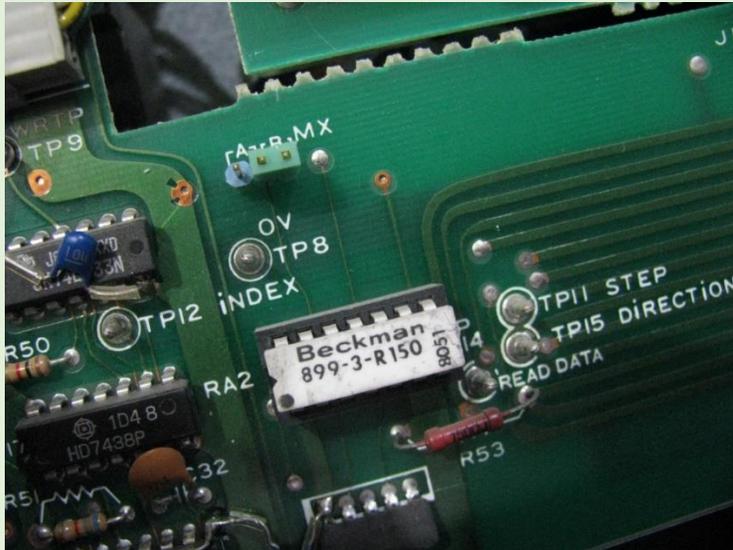
After some thought, I concluded that the problem may lie in one of these four elements:

1. The media (i.e. diskette)
2. The floppy drive
3. The cable
4. The controller

It was easy enough to test the media. I located another diskette and repeated the tests. It behaved identically. There were more comments on the Facebook group:

"I wonder if you have a bad cable? or upside down on both ends.

"Does your connectors have keyway in them?" - Mark Bosley

"If the cable is reversed the LED stays on.

"FYI: You can also hook a 3.5 inch 720k (not 1.44Mb though) drive to the CoCo, FYI, with the right cable. If the drive steps through all 35 tracks on a DSKINI, it is mostly working..." – Barry Nelson

"Salvador Garcia I seriously doubt the uprocessor is the issue. If it's not a bad cable the drive may just need cleaning, alignment, or it's speed adjusted." – Barry Nelson

I was afraid of this. I did not have an oscilloscope nor the skill to align the head. I didn't even have a florescent light to adjust the speed. I noted this on the Facebook group.

"For speed adjustment, many drives have a black and white disk on the hub that should appear stationary when illuminated by a florescent light if the drive is properly adjusted. Rotation speed might be a problem, but thinking

about it, alignment is probably not the issue. Bad alignment would have prevented the drive from reading other drive's disks, but I would still read its own" – Barry Nelson

Yup, check on the black and white disk on the drive. Unfortunately, no florescent light.

Head Cleanup

The last attempt I decided to make was cleaning the disk drive's heads, or in this case, head, since it was a one-sided drive. I removed the electronics board to access the drive's innards and located the head.

Yup, that's the head, the thingy that looks like a "pause" symbol. Carefully, with a Q-tip and some alcohol, I cleaned the head. After that was done, I reassembled the floppy drive. For me, that was the last activity that I could perform on the drive. I hooked everything up and retested. No change.

The Next Step

I recapped what I had done so far:

a. Cleaned the card edge connectors of the controller and drive.

b. Reseated the socketed chips on the controller and CoCo 1.
c. Located pin 1 on the drive and controller and correctly connected the 34-pin flat cable with the different colored line towards pin 1.
d. Used different media in case the problem was the diskette.
e. Cleaned the drive's head.

I decided to take the 26-3022 controller back to Chicago. The drive was bulky and heavy and would be hard to transport. My objective was to test the floppy disk subsystem by swapping out components.

I ordered the flat cable that I located on eBay. Next, I needed to locate a replacement for the floppy drive. Jason Downs recommended a GoTek drive. I had heard of it, but was not familiar with it. I did some research and found it to be an interesting device. The GoTek is a floppy drive simulator that has a standard 3.5" drive header connector. One of the members in the CoCo List,

Henry Rietveld, has posted videos using a GoTek with the CoCo. I decided to get one of these to connect to the 26-3022 controller.

The next step is to get a Gotek drive working with the new cable then once that happens, I will swap out the flat cable for the original one to determine whether this is the problem. I plan to connect the Gotek to both connectors, knowing that one responds to drive 0 and the other to drive 1.

Stay tuned, same CoCo channel, same CoCo time, for the next part of this adventure.

The Papoose Theorem
(Lest you ever forget)

by Bob Swoger

The Chief had three squaws. The first squaw slept on a moose hide and gave birth to a papoose. The second squaw slept on a bear hide and she gave birth to a papoose. The third squaw slept on the hide of a hippopotamus and gave birth to twins. (Familiar?)

So, the Papoose Theorem states that the squaw on the hide of the hippopotamus is equal to the sum of the squaws on the other two hides.

Operating System Nine

By Rick Ulland

25 years ago, I was an active OS9 user, and almost knew what I was doing! Then life happened. Last year I dug the old CoCo3 out of storage, and this series documents my journey back to almost.

Starting with a dead hard drive, but there was a backup - 60 odd 720k floppy disks, also 25 years old. Besides that, a floppy drive CoCo3 with MPI, some floppies that still worked, and many boxes of CoCo stuff. And the community. I was amazed at the number of people still campaigning the CoCo, and new products like the CoCo SDC certainly made restoring my old system more likely. Finding OS9 still under active development made it certain.

Getting started was easy. Since the 90s, NitrOS9 has gained the ability to suss out various sized floppy drives all by itself, understands the SDC, and provides an empty hard drive image to stream my backup to. And the stream program preinstalled, come to think of it.

Some Time Later.

Most of the old floppies were actually good! and I soon (heh) had the old hard drive mounted as /h1, with a bootable NitrOS9 EOU image as /dd. This is recommended as an easy way to keep the OS current without accidentally smashing your data, so we'll keep this setup for a while.

The first task is to tweak system settings to my likings. Plural, because a CoCo simply doesn't have the room or speed to do everything/all-the-time. We must divide and conquer.

Divide what? OS9 already divides the boot into three datasets, a file for each.
    /dd/OS9Boot - merged glob of OS9 parts to load
    /dd/startup - list of things to do at startup
    /dd/SYS/env.file - system wide environment settings

Because small systems generally used a boot floppy, there wasn't a menu system. Copy that floppy! We could still do that, with a bunch of SDC images for drive 0 and the separate data image in drive 1. But you might not want to, for example OS upgrades would become chore to the nth.

To manage these files today, NitrOS9 provides the swapboot utility. You might have already used it to set up drivewire or an emulator. It easy to create your own options. Copy the big 3 files, adding a custom file extension to each to ID a set.

Start with your original boot, as soon as it is semi-tolerable. Then experiment away, knowing the original is still close at hand.

```
cp /dd/OS9Boot /dd/OS9Boot.nos9
cp /dd/startup /dd/startup.nos9
cp /dd/sys/env.file /dd/sys/env.file.nos9
```

cp will yell at you if the extension (set name) is already used. To deliberately modify an existing set, add the overwrite option

```
cp -ov /dd/OS9Boot /dd/OS9Boot.nos9...
```

And remember, if the system ends up unbootable, you can mount it as a data drive, boot with any NitrOS9 image and manually cp -ov a set of known good files over the failed ones and try again.

The Unix Way.

You might think all this manual copying sounds a little old fashioned, and it is. OS9 has been described as Unix like, and here is an example of the Unix motto "Do one thing and do it well". Swapboot allows a casual user to change boot files from a menu. It could be expanded to look like an application, adding code to validate the user as administrator, perform crash recovery, and add things like overwrite confirmation dialogs. But cp already does all that, it's always available and well tested.

There used to be a second motto, "Keep it Simple Stupid". But we have to reach back to a time before XML, before the windows registry hive, back to the 8-bit ASCII text file.

Understood by man and machine, changeable with standard utility editors, and access controlled with file permissions.

Sometimes there will be a fancy UI tool to edit these files, like MultiVue's 'control' applet. But one can always cut to the chase and wade in there with a text editor.

Let's do that. /dd/SYS/env.file describes how you want your hardware used. Control manages some of this, in fact it's the best thing for setting colors. But first, let's swoop in there and set a few things...

MONTYPE = 0 for composite, 1 for RGB, 2 for monochrome

CONVTYP = text console resolution - 1 for 40x25, 2 for 80x25

CONXSIZ = text console width - usually 40 or 80 matching resolution

DEFTYPE = gshell startup screen resolution

    6 for 320x200x4

    7 for 640x200x4

    8 for 320x200x16

PTRRES = 0 for low-res joystick/mouse, 1 for hires mouse adapter

RBFDEV = disk drives or equiv - /dd /d0 /d1 /h1 /r1 etc

SCRDEV = printers and serial ports - /p /t1 /t2 etc

DATA = default data directory (usually /dd)

EXEC = default execution directory (usually /dd/CMDS)

To go further, check the file ENV_FILE_DOCUMENTATION.pdf for a complete list of tweaks you can make.

The Process.

In addition to configuration data, you might want to change the things that load or run at startup. These are listed (surprise!) in the file startup. NitrOS9 makes this pretty easy to edit, starting with all the things. Just take out the bits you don't need.

All you really need to do is link the command interpreter shell, to make sure this handy gadget doesn't get dropped on the floor. But the time prompt is easy to skip, and some extra windows are cheap to own. A basic text startup is simply:

```
link shell
setime<>>>/1
shell i=/w1&
shell i=/w2&
```

Adding graphics doesn't have to be painful. It's true all those fonts and patterns take time to load, but you don't have to load then all! At the rock bottom, you can run 4 color Multivue by adding this one line:

```
cd sys; merge stdfonts stdpats_4 stdptrs
```

This sort of quick loading boot is good for file wrangling, organizing MultiVue, and getting right to tasks like text entry.

```
cp /dd/startup /dd/startup.fmgr
cp /dd/OS9Boot /dd/OS9Boot.fmgr
cp /dd/SYS/env.file /dd/SYS/env.file.fmgr
```

Between these two extremes is the best boot speed for the apps you normally use. I'm sure there is a doc but things are getting a little long now, so...

That leaves the actual glob of modules in OS9Boot. In the past, this was part of initial setup. NitrOS9 has automated or relocated many of the things we used to change in this file, so it's not time to worry about it yet

But that time will come. My guess is next issue. Till then, see you on the online.

-ricku

LogiCall Review

LogiCall Review for CoCoCoding Contest 2013
By Kip Koon, The ComputerDoc

I have been on the CoCo List for about 2 years now gaining much knowledge about all the changes that have taken place since I left the CoCo behind in the middle of the 1980s. When the invitation was given to everyone to review a program entry for the CoCoCoding Contest arose, I thought for a little while about it not sure if I wanted to review any programs since they were mostly games. When I saw there was also a disk utility entry, I really wanted to review that one, so I jumped at the chance to give back to the CoCo Community. When asked what type of program I would like to review, I responded that I would like to review a utility program of some type. Not being sure what utilities were left, if any. I was surprised when I was offered *LogiCall*! I was very glad to get that program to review.

Before I begin the review, I thought I would share with you the history and origin of *LogiCall* as it was passed to me from Bob Swoger by way of John Mark Mobley.

History (by: Bob Swoger):

Back in the early 80's, a Hoffman Estates, IL based firm, Microsystems Engineering Corporation (MEC), won a bid to provide NASA with a computer automated testing package using Fluke test equipment. When the project was finished, they turned to making an integrated software package for the Digital Equipment Corporation (DEC) PDP-11 minicomputer series used by big business. Argon and Sandia National Labs, Dow Chemical, Coopers & Lybrand, Goldman Sacks, Northern Trust, Shell Oil, Sun Microsystems, Texas Instrument, Dupont, Edwards AFB, and MOTOROLA Research were some of many worldwide users. Their user groups, Beta testers and steering committees looked like a Who's Who of Fortune 500 companies. MEC's product was called MASS-11. You didn't buy this software, you leased it. The MASS part stands for Management Administration System Software and the 11 part stands for the computer series it ran on, the DEC PDP-11. MASS-11 contained a word processor, spreadsheet, database, DRAW application and modem package along with a disk management system. Any data created could be seamlessly moved from one application to another. Later, MASS-11 was made to run on the DEC Rainbow Desk Top computer. When the IBM XT, AT and DEC VAX entered the computing arena, MASS-11 was again updated to run on those machines keystroke for keystroke, a great selling feature. Remember, Hard Drives back then were only 10 Megabytes! MASS-11 was menu driven requiring only 3 keystrokes to launch and manipulate any application. Microsystems Engineering and MASS-11 still exist to this day, but both original owners, Hank and Everett Karels, have passed.

Hank Karels lived two doors from me. My wife went to work for MEC - so I became familiar with the MASS-11 product. It was the high-level human interface that interested me. Having at that time a $99 ZX81 computer, I felt that if MASS-11 could Call their applications using three keystrokes, I could do it on an 8-byte microcomputer using only one keystroke. *LogiCall* was born. In 1984 *LogiCall* moved from the ZX81 to the TS2068 and Spectrum computers, all being Sinclair machines. Using the LarKen Disk Interface *LogiCall* could run on either machine even though they didn't use the same firmware ROMs. *LogiCall* could tell which machine it was on and switch systems on the fly. I sold it in the continental US and Canada. In 1995 *LogiCall* was released to two vendors, Frank Davis in Mexico, Indiana and Rod Gowen in Oregon City, Oregon. Now John Mark Mobley and I have migrated *LogiCall* to the CoCo. Chris Hawks jumped in many times when we hit a brick wall.

More History by John Mark Mobley

A couple of years ago, Phil Kwitkowski, a friend of Bob Swoger, wanted to give away his Timex Sinclair 2068 system, so Bob arranged a time for me to come and pick up Phil's system. Since *LogiCall* ran on the Timex Sinclair, Bob taught me how to use both the Timex Sinclair 2068 and *LogiCall*. Bob had plans for many years to rewrite *LogiCall* to run on the CoCo, so we began working on this project. Bob would develop the code to work on Vcc, and I would test the code on real hardware. Bob and I connected by Skype, and I could show him how the code ran on my CoCo. Christopher Hawks helped us get a directory listing with two files per line, and helped us with setting or detecting caps lock. John Mark Mobley

More History by Christopher Hawks

I merely supplied the technical know-how. If Bob or John Mark needed to know how to make BASIC do something or where something was, I could supply that knowledge. (Change keyboard entry to upper/lower case, POKE 263,255 for upper, POKE 263,0 for lower. Sometimes programs run by *LogiCall* would change the case to lower and *LogiCall* needed upper case entry to continue running.) (And I did a lot of testing.)

Christopher R. Hawks, HAWKSoft

After another phone call with Bob Swoger, I have some more historical information. The following are notes I took of Bob's comments from that conversation:

"The Spectrum computer has a syntax checker. A beginning programmer could not make a mistake writing a BASIC program that would cause a Syntax Error at run time.

Here is a sample Sinclair BASIC command line to save a file.

RANDOMIZE USER 100: SAVE "filename.ext" LINE 600 <ENTER> - That's a lot of key strokes. Wouldn't pressing <S> save a lot of typing? That was the reasoning behind *LogiCall*'s creation. Necessity is the mother of invention, Laziness is the father. Therefore *LogiCall* was born.

As a young engineer, I began searching for lost endings of old sayings.

"Beauty is only skin deep, but ugly is to the bone." (Red Foxx)

"Necessity is the mother of invention, Laziness is the father. "

"There's more than one way to skin a cat, but only one way so as the holes don't show."

Engineering Quotes:

"There is nothing more permanent than a temporary installation." (Robert Loving)

"The pressure on an object applied by a soldering iron is inversely proportional to the heat of the iron." (George Hawkins)"

Thus endeth the history lesson. I thought that it was very interesting to find out about how *LogiCall* came to be created. I'm always interested in what spurred on the ideas and beginnings of computer products both hardware and software. Now on to my experience learning about, using and ultimately writing a review of *LogiCall*.

This version of *LogiCall* has been in the making for 18 years since it was first shown at a Glenside Color Computer Club meeting. *LogiCall* is actually an ensemble of many programs. L.BAS is the main program. John Mark Mobley wrote MOVE and FORMAT for both CoCo and DriveWire. Chris Hawks would fix any problems that arose. If it couldn't be found in the manual, they'd ask Chris. Function routines and a POKE to change keyboard to all caps are but few of the many things Chris helped with. Chris would write routines for the functions that Bob needed to replace Sinclair keywords missing in CoCo BASIC. Bob didn't compile *LogiCall* into machine code because BASIC is just the right speed to run *LogiCall*. Also, he wanted to allow the user to make modifications to the programs on his own.

My Introduction to *LogiCall*

As soon as I downloaded *LogiCall*, I began to read its documentation file (Manual). I really don't know why, but after reading the entire Manual, I could not wrap my head around how to use *LogiCall* at all and understand how the program worked. Now normally I'm very intuitive when it comes to figuring out software, but this time, I was stumped. I could not get it to boot up on XRoar with the Becker port activated no matter what I did. I had DriveWire 4 Server running so it should have come up running but it didn't.

So I sent out an email to all three programmers who had a hand in its creation including the original author who got back with me with very helpful responses. Thank you, guys. The original author of *LogiCall* for the TIMEX/Spectrum TS2068 computer, Bob Swoger, wanted to call me to help me through understanding the program, so I gave him my phone number and we chatted for quite a while about his background and how he came to create *LogiCall*. I wish I had a recording of that conversation. Bob is a very interesting man to talk to. Since I did not record the conversation, I will just include pertinent parts of our conversation to the best of my memory as we go through this review.

Before our conversation, there were so many files in the *LogiCall* zip file to figure out that I did not know where to begin. So now that I have received a guided tour of *LogiCall* from the original Master Programmer of the Spectrum version, I will attempt to pass on to you all the information that Bob Swoger gave to me.

First we went through the setup for Vcc.

1. The MultiPak Interface (MPI) must be inserted into the cartridge port slot.
2. The MPI Slot 4 must have the FD-502 Floppy disk controller cartridge pak installed.
3. The MPI Slot 3 must have the Hard Drive plus the Cloud9 RTC (Real Time Clock) installed.
4. The MPI Slot 2 must have the Orchestra-90 Cartridge pak installed.
5. The MPI Slot 1 can be left empty.
6. FD-502 Drive 0 must have the disk image BasicA.DSK inserted.
7. FD-502 Drive 1 must have the disk image BasicB.DSK inserted.
8. FD-502 Drive 2 must have the disk image BasicC.DSK inserted.
9. FD-502 Drive 3 must have the disk image BasicD.DSK inserted.
10. The FD-502 Configuration must have:
    a. The RGB DOS must be selected.
    b. Persistent Disk Images must be checked.
    c. The Clock at 0xFF50-51 must be checked.
11. The HD Drive should have the NitrOS-9.vhd file inserted.
12. Under menu item "Configuration > Config > Vcc Options > CPU tab", these options should be selected.
    a. 8192KB of RAM
    b. A 1.788 MHZ clock rate
    c. The Hitachi HD6309 CPU
13. Under the Display tab the RGB option should be selected.
14. Allow Resize should be checked.
15. Under the Keyboard tab > Keyboard Mapping pull down menu, select Basic to allow the <F1> key to work. The Keyboard layout of Vcc is changed from the original CoCo layout to match as much as possible the layout of the typical IBM PC compatible keyboard of today.
16. All other settings can use their respective defaults.

Now power cycle your virtual CoCo 3 and *LogiCall* should load and run automatically. The first file saved to the disk determines the disk's name, so don't save L.BAS (the main *LogiCall* program) to the floppy disk first. Instead, install the program with the name you wish to also name the disk. The following picture is the initial boot screen.

```
Disk  Name:  DISKS
%%%  for  Tandy  Color  Computer  V7.0   %%%

DISKS       BAS       SLOWPOKE  BAS
FASTPOKE  BAS         SETSPEED  BAS
FINDFILE  BAS         VIDEORAM  BAS
TERMINAL  BAS         NIBSHOW   BAS
TAPES       BAS       MOVE        BAS
COUNT       BAS       DWMOVE      BAS
AUTOEXEC  BAS         FORMAT      BAS
DWFORMAT  BAS         WRITER      BAS
CALC        BAS       ATTR        BAS
STEPRATE  BAS         L           BAS
RADIOH      BAS       STARTOS9  BAS
DRIVE=3    FREE=3




Drive? _
```

I really like the rich blue color for the background. Bob said that long ago people had discovered that this scheme was the most comfortable for your eyes.

LogiCall first asks what drive you wish to see. Once you enter the drive number *LogiCall* puts that disk's CATalog on the screen and next asks you for a program to run (or command to execute). More on this later. You will immediately notice that *LogiCall* booted to Drive 3. Bob said that he liked coming up on Drive 3 initially so as to not accidentally corrupt important files on Drive 0, his *LogiCall* install disk. After using *LogiCall* for a few days, I think I am beginning to see why. Most important programs tend to like to live in Drive 0, so having *LogiCall* boot to Drive 3 makes sense to me. Upon booting the CoCo, the AUTOEXEC.BAS file on VHD Drive 0 executes the high-speed POKE, installs the 4 virtual floppy drives and sets the system pointer to virtual floppy Drive 3.

LogiCall itself is only 3 granules, so it can be put on just about any disk you like. Bob explained that *LogiCall* (L.BAS) should be saved on each and every populated disk. Again, since the disk name is picked up from the first file in the CATalog, don't save L.BAS to the disk until the main program is first installed. *LogiCall* always comes up listing just BASIC programs on the current drive, so if there is a Binary file or any other file type for that matter, just press the <V> key, which stands for Verbose, and all files on the disk will be displayed. Press the <B> key to return the brief CATalog.

Most of the "Program?" prompt commands also work at the "Drive?" prompt. When I first started using *LogiCall*, I thought I could only use drive numbers. Not true! I can use just about all the single letter commands also. "T" is one exception. At the "Drive?" prompt "T" calls the Tape Drive. At the "Program?" prompt "T" calls the Terminal program.
At the "Drive?" prompt, to change to another drive, just press the <0>, <1>, <2>, <3> or <T> keys and the system pointer will point to that drive and display its contents or in the case of <T> will list the programs on the tape. In fact, all the controls for this program are only single keystrokes! One key with <ENTER> and that function takes place immediately! To display the programs on a cassette tape, just press <T> and the "Play Tape" prompt appears in the middle of the screen. The program then waits for you to begin playing a program or data tape on the cassette recorder.

If the numbers 1 through 4 or 10 through 255 are entered at the "Program?" prompt, the system pointer points to that drive and its CATalog is displayed. The keyboard numbers 5 through 0 were used in the *LogiCall* for the Sinclair long before a CoCo version was ever developed.

```
                 L o g i C a l l
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        You may select drives T through 3

            At the Program? prompt:
    ENTER & space scans CATalog for LOAD

        5      RENAMES oldfile, newfile.
        6      Calls in the MOVE Program.
        7      ERASES file from disk.
        8      Returns the Drive? prompt.
        9      Refreshes the CATalog.
        0      Calls in the FORMAT program.
        R      RENAMES oldfile, newfile.
        S      Saves this program to disk.
        T      Calls in the Terminal program.
        W      Calls in the Word Processor.
        Z      COPYs CATalog to printer.

        Press <ENTER> at the Program? prompt
        and use the arrow keys to select a
            program to LOAD and RUN.

             Press <ENTER> to CONTINUE_
```

One of the first functions Bob showed me was the Help Screen, there are 2. The first Help screen is shown above. You get the Help screen using <H>, <?> or <F1> and <ENTER>.

I immediately saw something strange. The key <7> erases a filename. Huh? So I asked Bob, "Why? That doesn't make any sense." He said that the Timex TS2058 and Spectrum computers have all the BASIC commands printed on every single key of the keyboard and ERASE is printed on the <7> key. *LogiCall* is keystroke for keystroke compatible on both the Sinclair and the CoCo as was MASS-11 on all the machines it ran on. He said it was impossible for a beginner to miskey any command, because the Sinclair had a syntax checker that wouldn't allow the user to enter a bad line of code. Of course, having all the keywords printed on the key caps helped the new programmer and saved extra typing too. In fact, every key on a Sinclair has five or six separate functions! I said, "Oh, that makes a lot more sense." So, he set out to show me what every single letter command in the HELP file did in *LogiCall*.

RENAME

So, to rename a file on the current disk press the <5> key or <R>, and the "RENAME Old?" prompt appears. Enter the old filename, press <ENTER> and the "New?" prompt appears. Enter the new filename and press <ENTER> to complete the operation. (He had me rename L.BAS to L2.BAS)

ERASE

At the "Program?" prompt, pressing the <7> key or <E> will display the "ERASE?" prompt. Entering a filename and pressing <ENTER> will immediately erase that program file from the current disk. (He had me erase L2.BAS. Wow *LogiCall* was no longer in the disk. What is he doing?)

Saving *LogiCall*

At the Program? prompt pressing <S> will save the *LogiCall* program itself to the current disk with the filename "L.BAS". (L.BAS reappeared in the CATalog. The disk was now back in the state it was in before we called RENAME. WOW)

Drive?

If you press <ENTER> at the "Drive?" prompt like I accidentally do so many times, pressing <8> or <D> at the "Program?" prompt will return the "Drive?" prompt instead of having to execute a program just to get back to the "Drive?" prompt. You can also just press the <End> key in Vcc or <Q> or the <Break> key on a real CoCo to halt the program, then just enter the "RUN" command to begin *LogiCall* once again to return the "Drive?" prompt. Typing ON or OFF 3 will turn the real or virtual floppies off or on when using Vcc or DriveWire. (*The disk name will not update until you change drives*.)

CATalog

At the "Program?" prompt pressing <9> or <V> displays the CATalog or directory of the current disk, so basically it refreshes the CATalog display.

MOVE

At the "Program?" prompt, pressing the <6> key or <M> calls in the MOVE program. The Move Utility gets loaded into BASIC's program memory replacing *LogiCall*. Within the Move Utility program, you can move a single file from one drive to another drive by entering <1> or <M>, rename a file on a disk by entering <2> or <R>, and save the Move Utility to disk by entering <3> or <S>. To exit back to *LogiCall*, enter <0> or <E> and *LogiCall* is loaded back into BASIC's program memory. Here is the MOVE & RENAME Utility screen.

```
LogiCall  MOVE  &  RENAME  utility    V1.5
        #########
        #LogiCall#
        ]###  ####
        ###   ###
        ####  ####
        #########
        ####[]####
        ####[]####       Select One
        ####[]####
        #########
   '0'    Exit  MOVE

   '1'    MOVE  a  single  file

   '2'    RENAME  a  file  on  disk

   '3'    SAVE  this  utility  to  disk

Enter  0,  1,  2  or  3_
```

FORMAT

At the "Program?" prompt pressing <0> or <F> calls into BASIC's program memory the FORMAT program replacing *LogiCall* just like the MOVE program did. You can format a new disk by pressing <1> or <F>; to copy an entire disk to another disk press <2> or <C>; save the FORMAT program to the current disk press <3> or <S>. To return back to *LogiCall* press <0> or <E> and *LogiCall* is reloaded back into BASIC's program memory. The FORMAT & COPY Utility screen is shown below.

```
LogiCall FORMAT & COPY utility   V1.4
                        #########
                        #LogiCall#
                        ]###  ####
                        ###    ###
                        ####  ####
Select One              #########
                        ####[]####
                        ####[]####
                        ####[]####
                        #########

Exit FORMAT             '0'
FORMAT a New Disk       '1'
COPY entire  Disk       '2'
SAVE this program       '3'
to Disk

Enter  0, 1, 2 or 3_
```

How might you break into the program to examine the code? Press <4> or <Q> and the program stops. LIST will list the program. "RUN" will start FORMAT.BAS again.

Did you really mean to call the MOVE program instead of the FORMAT program? Press <M> and the MOVE program is called. Press "F" and the FORMAT program is called back in.

DWMOVE and DWFORMAT are the DriveWire versions of the MOVE and FORMAT utilities. *LogiCall* determines which machine it is running on and calls the proper utility to perform these tasks without the user having to do anything different for each system on which *LogiCall* is running.

Added Single Letter Calls (by Bob Swoger)

"Back in the days when I started writing and using *LogiCall* I found that I was using mostly my Terminal Package and modem to contact my favorite BBS or I was using my word processor and printer to compose documents including Newsletters and papers for work. I therefore set up <T> to call my Terminal package and <W> to call my Word Processor."

Terminal

Only at the "Program?" prompt, pressing <T> will load the Terminal application launching program into the BASIC's program memory replacing *LogiCall* just as before. A picture of the Terminal Application display screen appears below.

```
        Terminal Application  V0.4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



Select One

EXIT                          '0'

Run Terminal application  '1'

Save downloads to Disk    '2'

Save this program         '3'
   to Disk

   Enter  0, 1, 2 or 3
_
```

To run a terminal application, press <1> or <R> and the terminal application will load into memory. *LogiCall* defaults to loading "V-TERM Terminal Emulator". To SAVE downloads to Disk press <2>; To save this terminal application launching program to disk press <3>. To exit V-TERM, press <Alt> and <Q> together or if QUIT is highlighted with a black background press <Alt> and <ENTER>.

Word Processing

At the "Program?" prompt, pressing <W> will call the Word Processing Application launching program into memory. Its screen is shown below.

```
           Word Processor Application
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx




   Select One
   Return to LogiCall     '0'
   Launch Word Processor  '1'
   Save document to Disk  '2'
   Save this program      '3'
     to Disk

      Enter  0, 1, 2 or 3
_
```

Pressing <1> will launch the Simply Better program into memory. <2> will save a document to disk and <3> saves the Word Processing Application Launching program to the current disk. The Simply Better Boot screen is shown below.

```
<< Boot for SIMPLY BETTER program >>


Key <ENTER> to boot SIMPLY BETTER using
its default data module, or...
Enter the NAME.EXT of the module to use
instead, or...
Key a '?' and <ENTER> for DIR, or...
Key <BREAK> to abort...

         ->
```

Pressing <ENTER> will boot Simply Better with its default data module or enter the NAME.EXT of the data module to use, or press <?> and <ENTER> for a directory. A directory is shown below. You will notice the directory of drive 50 is being shown.

```
SMPLYBTR  BAS  1  SIMPLY    BIN  12
PART2     BIN  2  CONFIG    BAS   1
C         BIN  3  HELPFILE  TXT   1
POCKET    CRD  4  DFALT     BIN   1
RGB128    BIN  1  AMB128    BIN   1
GRN128    BIN  1  CREAT2*0  BAS   2
SIMPLY    BAS  1  DMP110    512   1
DOS_BOOT  DAT  1  DMP105    512   1
BOOT      BAS  1  AUTOEXEC  BAS   1
SIMPLY    DOS  1  BETTER    BIN   2
L         BAS  3  JOKE      TXT   1
WRITER    BAS  1
DRIVE=50     FREE=24

Press <ENTER> to continue...
```
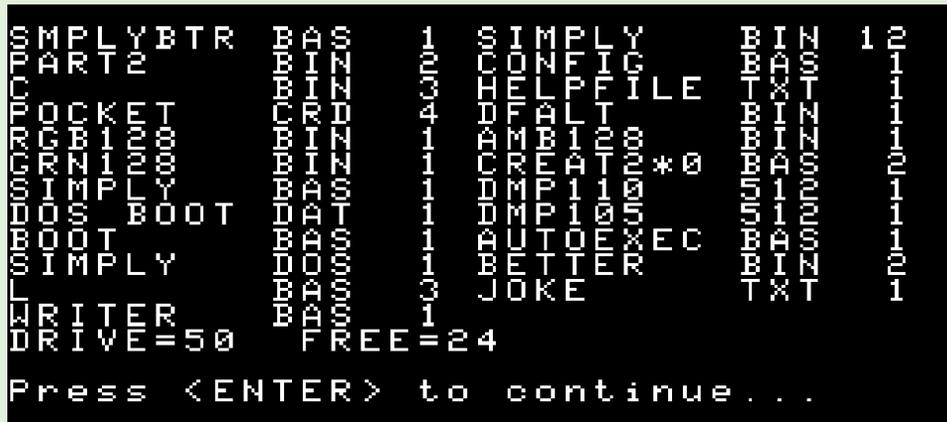
Pressing <ENTER> returns to the Simply Better Boot screen. And finally, pressing the <Break> key aborts. Entering RUN"L returns to *LogiCall*. Unfortunately, I do not know the difference in the different data modules beyond what the filenames imply. If there is ever an opportunity in the future to expand this review to include more detail, the differences in these data modules can be explored.

Printing the CATalog
Pressing the <Z> key displays "Printing…" at the bottom of the screen letting the user know the CATalog is being sent to the printer. Place the print-out in the disk sleeve w/disk for future reference.

Adding (Creating) AUTOEXEC.BAS
Before *LogiCall* the user had to create his own AUTOEXEC.BAS file. *LogiCall* practically writes it for you. Pressing the <A> key adds the "AUTOEXEC.BAS" file to the current disk. Typing "LIST" instead of "RUN" in the prompted AUTOEXEC creation sequence allows the user to modify the small AUTOEXEC program. After your modifications, type "RUN" to save the new AUTOEXEC.BAS program to the current drive.

Activate AUTOEXEC.BAS
If AUTOEXEC.BAS is on the current drive, pressing the <N> Key for NEW activates the AUTOEXEC program which amoung other things re-launches *LogiCall* as if you had just powered on your CoCo or booted Vcc.

Brief CATalog
Pressing the <B> key displays a brief CATalog. In other words, only BASIC program files are listed. This is the default display mode of *LogiCall* when it first boots. Most programs written in machine code have a BASIC driver that calls a screen file and then the machine code file. It is the BASIC driver that the user can modify to provide a logical return to *LogiCall*.

Verbose CATalog
Pressing the <V> key displays a verbose CATalog. In other words, all program files are listed. This is the default display mode of *LogiCall* when it first boots.

Second Help Screen
Now Bob went on to explain the second Help screen. It is shown below.


```
              LogiCall
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       You may select drives T through 3

          At the Program? prompt:
    ENTER & space scans CATalog for LOAD

     A   Adds AUTOEXEC to disk.
     B   Displays a Brief CATalog.
     C   Copies CAT to large printer.
     F   Calls in the FORMAT program.
     H   Calls in this HELP file.
     K   Saves #creen to disk.
     M   Calls in the MOVE program.
     N   NEW, activates AUTOEXEC.
     Q   Quit, halts this program.
     V   Displays a Verbose CATalog.
     Y   Returns to TERMINAL mode.

     During the AUTOEXEC sequence typing
     LIST instead of RUN allows the user
     to modify the AUTOEXEC program.

          Press <ENTER> to CONTINUE_
```

As you can see, many of the commands have more than one keystroke choice to activate that particular function giving you logical choices and thus the name of the program was born. Cool, huh! I thought that was a very logical name for the program. ☺

Quit
Pressing <Q> and <ENTER> will return you to Disk Extended Color BASIC's "OK" prompt. An interesting side note. In the early days of Microsoft when its owner, Bill Gates, was programming the initial Color BASIC 1.0 interpretive language ROM for Tandy, he changed the "Ready" prompt that the BASIC language had always used up to that point in time to the now familiar "OK" prompt to save three more bytes of precious ROM space. Bill did everything he could to reduce the ROM memory space required to store Color BASIC 1.0 in order to add as many functions as possible thus keeping the COLOR BASIC interpreter as compact and efficient as possible. That of course is definitely not the case today with Microsoft's software packages being so huge as we all know so well.

Return to Terminal
Pressing the <Y> key will return you to the Terminal program.
NITROS-9
LogiCall can boot NITROS-9 by first going to the NITROS-9 disk (255) and typing "DOS" at the Program? Prompt.

Alternate Program Loading
At the Program? prompt you can type in the program name with or without the extension from the disk CATalog or you can press <ENTER> at the Program? prompt and the first program on the list will become highlighted and you can then use the arrow keys to move the highlight to any desired program. Press <ENTER> again and the program will load and execute.

STEPRATE.BAS
STEPRATE.BAS allows you to change the head step rate of your floppy drives. When I run STEPRATE on my particular Vcc installation it shows the following screen.

```
YOUR  CURRENT  STEP  RATE  =  6  MS

   1  CHANGE  STEP  RATE  TO  30  MS
   2  CHANGE  STEP  RATE  TO  20  MS
   3  CHANGE  STEP  RATE  TO   6  MS
   4  SAVE  THIS  PROGRAM  TO  DISK
      PRESS  <ENTER>  TO  EXIT


      ENTER  SELECTION?  █
```

As you can see, pressing <1> changes the step rate to 30 milliseconds, <2> changes the step rate to 20 milliseconds, and finally <3> changes the step rate to 6 milliseconds. Pressing <4> saves the program to the current disk. Pressing enter exits the program and returns to *LogiCall*. Pretty simple! Huh! ☺

Summary (by Bob Swoger)
Ever hear your floppy drive chatter when loading in a program? This is because Floppy Drive Controllers are programmed for a 30 ms head step rate. The later drives could handle 6 ms or less for faster load time. STEPRATE.BAS allows the user to change the head step rate to match the drive. The result is NO MORE DRIVE CHATTER and faster read and write speed.

DISKS.BAS
The DISKS.BAS program is considerably more complex. Think of this program as a file card box. It holds a catalog of 40 cards (BOXes of disks - 0 thru 39) with 10 disks per box with notes on the right. **Summary** (by Bob Swoger)
DISKS allows the user to peruse up to 400 program files to find their location on your now 256 sectors of your virtual hard drive. It was the need for this program with today's Vcc and DriveWire's VHD both with 256 drives that showed us at Glenside that there was a new need to locate programs. The TAPES version handles 800 tapes with no notes.

After creating a new disk you add the information about that disk in this file so that you can find it again in the future. Remember the old saying; "The job is not finished until the paperwork is done." DISKS was created to help you fulfill that need.

The initial screen showing Box 0 is shown below.
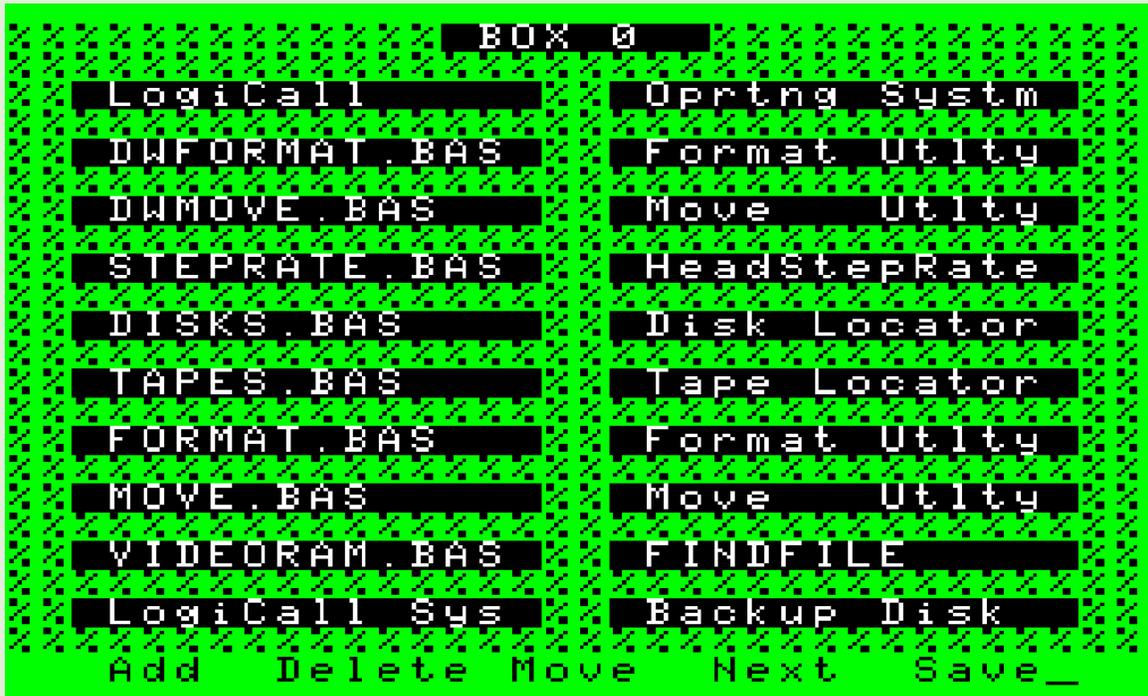
Entering a two digit number goes directly to that box number. There are some special cases though. Pressing <0>, <4>, <5>, <6>, <7>, <8> or <9> for box 0, 4, 5, 6, 7, 8 or 9 respectively is automatic, the box is immediately displayed. Pressing <1>, <2> or <3> requires the user to press <ENTER> afterwards. This is because 1, 2 or 3 is waiting to see if you want to press yet another number to call boxes 10 to 39.

<B>ack or <R>everse or <left-arrow> (Back to previous lower box number)
<N>ext or <F>orward or <Space> or <right-arrow> (Next higher box number)
Although the menu at the bottom of the screen does not show it, <B> or <R> or <left-arrow> go to the previous lower box number, and <N> or <F> or <Space> or <right-arrow> goes to the next higher box number. If you are at Box 39 and press <N> or <F> or <Space> or <right-arrow>, the program will go around to the beginning of the list and display Box 0! As you might expect if you are displaying Box 0 and press <B> or <R> or <left-arrow>, then Box 39 is displayed! The program does not bomb out! Good Work Guys! You thought of everything.
As an example of how to use this program, let us imagine you want to find V-Term, the Terminal program. From *LogiCall* on Disk 3, press <ENTER> three times. BOX 0 of DISKS is displayed, Press the <Space> bar until you see V-TERM (in BOX 7). The note to the right says the program is on Drive 49. Press <Q> and DRIVE 3 is displayed. Enter 49 at the Drive? Prompt and <ENTER>. Press <ENTER> three times and V-TERM is now running.
Now to "Do the Paperwork". Manipulating the information in these 40 boxes is what the short menu at the bottom of each is for. You can Add, Delete or Move information to any slot in the 40 boxes. It is a little tricky at first but gets easier the more you use it.

HELP Screen
<H> or <?> or <F1>
Pressing <H>, <?> or <F1> displays DISKS Help screen which is shown below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Each BOARD represents a File Box
   containing 20 SLOTS, 10 down the
   left side and 10 down the right,
   numbered 1 thru 20.  Add, Delete,
   or Move NAMES in the slots using
   the Select Keys shown below.

      Option          Select keys
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      ADD              A  +
      DELETE           D  -
      MOVE             M  6
      NEXT             N  F  Space
      BACK             B  R
      HELP             H  ?  F1
      PRINT            C  K  L  V
      COPY             P  Z
      SAVE             S
      UPDATE PNTR      U
      EXIT OR QUIT     E  Q

      PRESS  <ENTER>  TO CONTINUE_
```

As you can see there is sometimes more than one key that can be used to accomplish a function. Pressing <ENTER> takes you back to the Main DISKS display screen showing the current Box number.

nn (two digit box number 0 - 39) takes you to the desired box.

<A>dd or **<+>** (Add entry into a slot)
Pressing <A> brings up the prompt "ADD MODE – SLOT(1 – 20) ? " and then the user enters a slot number 1 through 20 and then presses <ENTER> indicating into which slot the user wishes to enter data. The DISKS program then displays the "TITLE?" prompt, then the user enters the title of the disk for that slot in the current box and presses enter.
<D>elete or **<->** or **<7>** or **<0>** (Delete entry in a slot)
Pressing <D>, <->, <7> or <0> enters the delete mode with the prompt "DELETE MODE – SLOT(1 – 20) ?". The user enters a number 1 – 20, then <ENTER> and the data in that slot is deleted.
<M>ove or **<6>** (Move slot data from a slot in the current box to a slot in another box (or the same box)
    Pressing <M> displays the prompt "MOVE FROM SLOT(1 – 20) ? ". The user enters a slot number and presses <ENTER>.
The prompt "DELETE FROM OLD SLOT (Y\N)" is displayed and the user responses with <Y> to delete the data in that slot or <N> or <ENTER> to leave the data unchanged.
Then the prompt "MOVE TO WHAT BOX (0 - 39)?" is displayed. The user enters the box number for the box the user wishes the data to be entered into and presses <ENTER>. If a number is not given *LogiCall* assumes you want the current box.
The prompt "MOVE INTO SLOT (1 – 20) ?" is displayed and the user enters the slot number to receive the data and the data is automatically entered into that slot.

Printing
<C> or **<K>** or **<L>** or **<V>**
    Pressing <C>, <K>, <L> or <V> will print the entire database from Box 0 – Box 39.
Copy
<P> or **<Z>**
    Pressing <P> or <Z> copies the screen to the printer. If using VCC, printing works by activating the bit-banger port and opening a text file for the printout to be stored. After the printout, the program returns back to the current box.
<U>pdate – Update Drive Pointer
    Pressing <U> will display "Drive?" in the middle of the screen. The user enters the desired drive number to change to in order to save another or separate copy of the program and database file on the new desired drive.
<S>ave
    Now last but definitely not least, pressing <S> displays "Saving on Drive 3" which saves all of DISKS database information and the DISKS.BAS program itself to the current drive which was the drive I was on when I started the DISKS.BAS program in the first place.

Don't forget to save all your data changes. If you quit DISKS.BAS without saving, then all your changes are gone, and you have to make your changes all over again!

<E>xit
Pressing <E> saves the entire database file and the program DISKS.BAS to the current drive and exits the DISKS program and returns to *LogiCall*.

<Q>uit
Pressing <Q> just quits the DISKS program without a save and returns to *LogiCall*. All changes are lost.

TAPES.BAS

TAPES.BAS operates exactly the same as the DISKS.BAS program since all the controls are exactly the same. The only difference is you store 20 tape names in the 20 slots in each box for a total of 800 tapes instead of 400 disks with notes.

This completes the review of all the commands of the *LogiCall*, STEPRATE, DISKS and TAPES programs. I will be using *LogiCall* quite bit in all my DECB only installations both emulators and real CoCos as much as possible. It really saves a lot of time! Of course, I will not be able use it for NitrOS-9 since it is written in Disk Extended Color Basic and cannot read NitrOS-9/OS-9 formatted disks. Of course, maybe the *LogiCall* programming team might create a NitrOS-9 version someday, but hey, this is definitely my own speculation. I do not have any knowledge of such an undertaking at this time, but who knows…

There is so much information in *LogiCall* to review. Since this has to be typeset tomorrow, I must put this to bed.

Further Information about *LogiCall* (given to me by John Mark Mobley from YouTube.)
The link for an excellent introductory video of *LogiCall* by Bob Swoger himself on YouTube follows along with the caption listed below the Video. The History portion has been integrated into the beginning of this review on Page 1.
https://youtu.be/f-R4UfU9_Cw

This is a glance at the *LogiCall* operating system as part of the Vintage Computer Festival Midwest Version 6.0 and Emergency Chicagoland Commodore Convention held on Sept. 24-25, 2011 in Lombard, IL. The computer used in the video is a Timex/Sinclair TS-2068 computer running LogiCall. *LogiCall* is an Operating System (written in Cambridge BASIC) that provides disk management and program selection from the disk catalog/directory. The program is written by Bob Swoger of --==GATOR==-- Software Development.

Here is a list of various 8-bit Microprocessor and retro computing platform conferences' and festivals' web sites.
Vintage Computer Festival: http://www.vintage.org/
Vintage Computer Festival Midwest: http://vcfmw.org/
Chicago Classic Computing: http://chiclassiccomp.org/
Southwest Regional Association of Programmers (SWRAP): http://swrap.org/
Suburban Chicago ATarians (SCAT): http://scatarians.org/
Glenside Color Computer Club Inc.: http://www.glensideccc.com/

Information on various computers mentioned in this review in no particular order:
http://en.wikipedia.org/wiki/Sinclair_ZX81

http://en.wikipedia.org/wiki/Timex_Sinclair_2068
http://en.wikipedia.org/wiki/ZX_Spectrum

For further information about the Timex/Sinclair TS-2068, access the following link.
http://en.wikipedia.org/wiki/Timex_Sinclair_2068

Many thanks go to Bob Swoger, John Mark Mobley, Christopher Hawks and Aaron Wolfe for all their help without which this review would not have been possible.

Personally, I think John Mark and Chris did a lot more than they let on in the histories they provided. I know Bob Swoger did for sure. Writing a comprehensive software package and testing that package requires a lot of work not to mention converting that software package to run on other sometimes dissimilar computing platforms. In order to create a cohesive software package like this much research, planning, coding and testing not to mention more research, more coding, and more testing go into a big project like this. The cycle continues until you feel you have included everything you wanted in your creation. Once you feel it's complete then you allow others to do alpha testing and beta testing until ultimately a final version of the software package is released hoping people will like it.

Check out this collection of programs for yourself. You just might be glad you did. I know I am. I hope this review has given you some idea of what *LogiCall*, STEPRATE, DISKS and TAPES can do for you. Happy *LogiCall*ing on your favorite computer – The Great and Mighty Color Computer 1, 2 & 3! Qaplah! StarFleet Out. End Transmission. ☺

Getting Started with the Game Master Cartridge

By Salvador Garcia

Introduction
The Game Master Cartridge (GMC) is a developer's kit produced by John Linville, now in its second version. The cartridge includes the electronics for a ROM with sound capabilities onboard. The GMC's objective is to enable developers to create cartridge-based software, such as games, without having to worry about what sound capabilities are available.

Once the software is done, the developer and John Linville discuss production and distribution. The developer needs to worry only about the software development, not the hardware, which is a major obstacle for many developers,
This document guides the reader on how to get started with the GMC in a Windows system by doing a step-by-step setup of the development environment and creating a BASIC and assembly language program that generates a simple sound from the GMC. Before I begin, I want to acknowledge the help received from Steve Bamford. He is currently working on a project for the GMC and I asked him for some notes and tips on getting started. Instead of notes, he provided a comprehensive document with lots of information. This article would be exponentially simpler if it weren't for these notes.

Assembly
John designed the GMC so that it would use through hole components, making it easy to solder them to the Printed Circuit Board (PCB). Version 2 includes an additional capacitor, deemed necessary for the sound chip used. Therefore, version 1 needs an engineering change which is the addition of the capacitor.
I bought my GMC back 2018 when the capacitor was not included in the design. When version 2 came out, John provided a detailed document on how to modify the original version. The existing resistor, on version 1, goes from the sound chip to ground. John determined that a capacitor in series needed to be added between the resistor and ground. The next image shows the resulting assembled GMC, including an insert that shows the hacked upgrade.

To prevent the CoCo from executing non-existing ROM code, I decided to jumper pins 7 (Q) and 8 (CART*). The image of the assembled GMC shows the header and jumper. This image also shows how I added the capacitor to bring the GMC to version 2.
Note that only version 1 needs this modification. If you bought the GMC recently, then the GMC that you have is version 2 and has separate spaces for the resistor and capacitor. The next image, provided by Steve, shows Steve's version 2 GMC. Note the reserved spaces for these two components (bottom right corner of the PCB).

While I chose to hack on a jumper and header for pins 7 and 8, Steve chose the venerable trick of putting tape on top of pin 7 to sever the connection to pin 8, as is highlighted by the red circle in the image.
Why is this needed? While there is no cartridge, a friendly pull-up resistor keeps CART* in a high state, telling the Coco

that there is no ROM code to execute. When a cartridge is inserted, the CART* pin is asserted low telling the CoCo that there is a cartridge with ROM code attached. The ROM code begins at address $C000 and when CART* is asserted, the CoCo jumps to this address where it expects to find valid code. If we just want to work with the sound chip and don't have any ROM code then we need the cartridge to still be functional but prevent the CoCo from executing non-existent ROM code.

We can do this by tricking the CoCo into believing that there is no ROM code by isolating the CART* line. Once we do have ROM code, we can reconnect the CART* to the Q line. Steve can remove the tape, or I can short the header pins with a jumper.

Emulation

One question that arises is: When I have code, do I need to burn an EPROM over and over again to try it out? Fortunately, the answer is no. Three of the most common CoCo emulators support the GMC: XRoar, MAME and VCC. What this means to developers is that we can develop and test our software in a virtual environment, without having to burn an EPROM every time we have a new version of our software.

This is a boon to productivity because can make changes galore to our code and testing it is much simpler, without needing real hardware, such as the EPROM or its corresponding programmer.

The documentation that John provides presents more information about how to use the XRoar and MAME emulators. Reading this is recommended, even if the document is an extremely long seven pages! 😊.

Setting up the dev environment: XRoar

The first task to getting started with the GMC is setting up the development environment. We'll start with the CoCo emulator XRoar. Steve chose to use this emulator, so this will be the one discussed in this text. As mentioned, both VCC and MAME have support for the GMC, so the developer is free to choose any of these.

To install XRoar, download the binary for the edition of Windows on the computer that was chosen for the development station. Go to the following link:
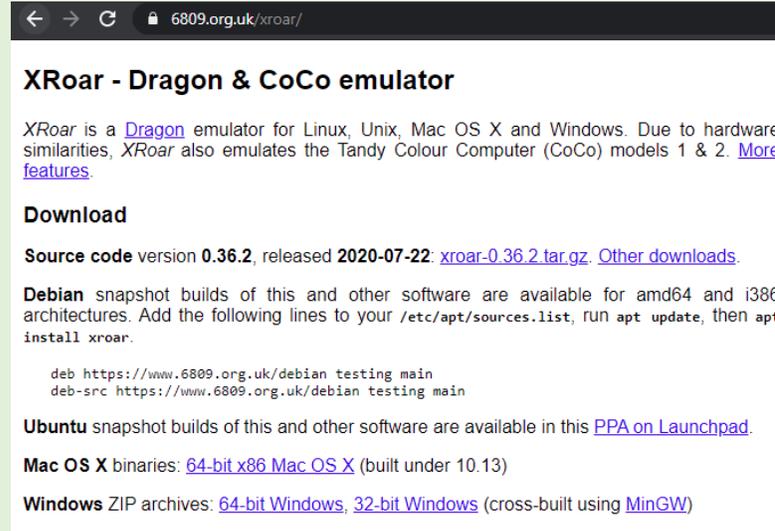
https://www.6809.org.uk/xroar/

Locate the Windows ZIP archives section and click on the corresponding binary for the dev station's edition of Windows (32 or 64 bit). Once the download is done open the ZIP and copy the xroar folder to a known location on the dev station. The next image shows the files that are originally in the XRoar folder.
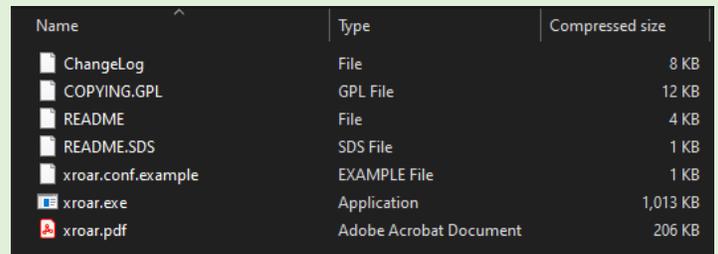
The "exe" file is the application executable, while the "PDF" is the user's guide. I recommend reading (or at least skimming through) the documentation.

We are not done though. XRoar, and other emulators, will work only if they have access to the Color Computer ROMs We need to get those ROM files and copy them to the folder where the XRoar executable lives.

Note: There are more elaborate ways of setting up XRoar. I am presenting the simplest. For more information, I refer you to XRoar's documentation.

The Color Computer Archive has a set of ROMs for XRoar. Go to the following link:

https://colorcomputerarchive.com/repo/ROMs/XRoar/CoCo/

You can download every .ROM file in BASIC_OS or pick and choose. The two .ROM files that I used were bas13.rom and extbas11.rom. For this exercise, no .ROM files from the DOS folder are needed. Put all the ROM files in the same folder where XRoar.exe lives. This is what my XRoar folder looks like:
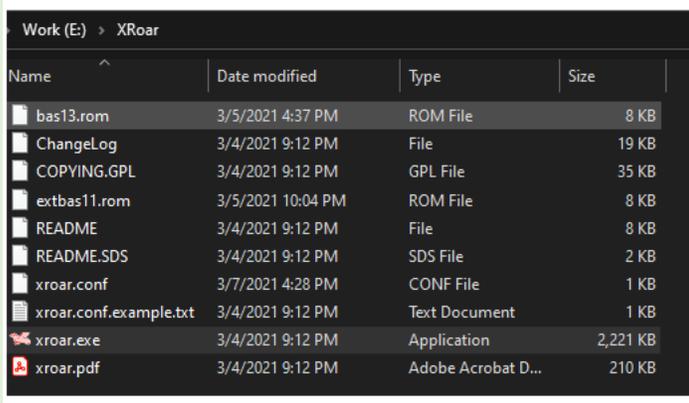
This emulation does not implement the Disk functionality. If this is desired, then XRoar needs to be setup differently. Read all about it in the XRoar documentation.

Double clicking the executable runs XRoar, as shown on the next image. The copyright and MEM available might vary, depending on which ROMs you decided to put in the folder.
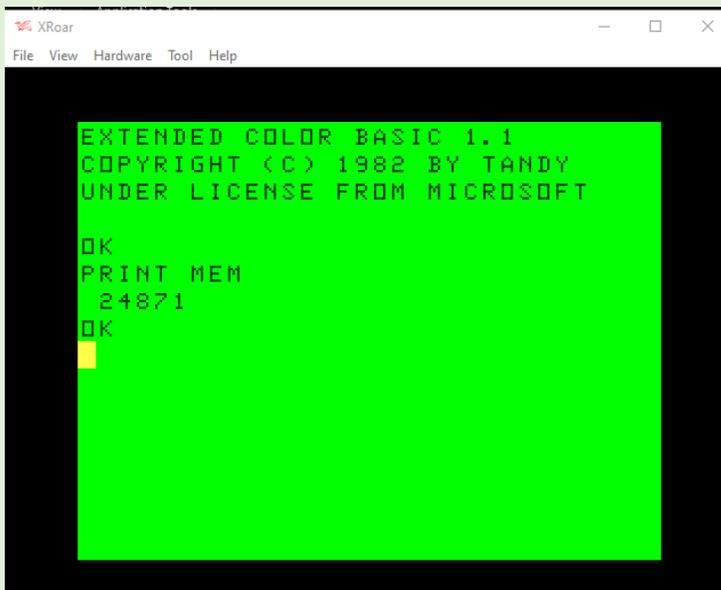
To make use of the GMC, we need to "plug it in" by selecting **Hardware** from the main menu, then **Cartridge** and lastly **Game Master Cartridge**
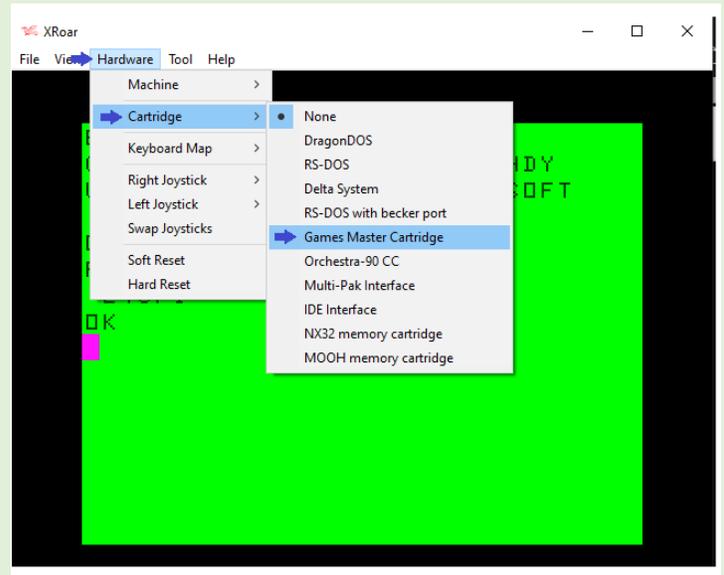
This setting is not persistent. When we run XRoar again later, we'll have to set up the GMC again. Let's fix that.

Open a text editor (such as Notepad or functionally equivalent) and enter or copy and paste the following text:

```
### Configuration file for XRoar for GMC
development
#    by Salvador Garcia

kbd-translate

# Define GMC cartridge
cart GMCDev
  cart-desc "GMC Development"
  cart-type gmc
  #cart-rom MyROMSoftware.rom
  no-cart-autorun

default-machine coco2bus
machine-cart GMCDev
```

Lines starting with a pound/hash sign (#) are comments and can be omitted. The comment with the command cart-rom allows us to use the GMC without any ROM code. Once we have some ROM code, we can remove the pound/hash sign to activate the line and comment out the no-cart-autorun line so that the ROM code will execute.

NotePad++

Next, we'll set up the editor that we can use to enter our program. Although there are many code editors available, Steven chose NotePad++. I too have used this editor in the past, so I chose it as well. To download it, go to the following link:

https://notepad-plus-plus.org/downloads/

It is a good idea to always select the most recent version which will be at the top of the list. Click on the link and it will take you to the download page. Click on the download button under the box.

Download 32-bit x86

Click anywhere inside here

Don't click here

~~Start~~

3 Easy Steps
1. Click "Start"
2. Start Download
3. Browse The Web

Wave Browser

Or here! ~~Start~~

This will download the npp.X.Y.Z.Installer.exe file, where X.Y.Z will be the version being downloaded. Run the installer and NotePad++ will be ready for use.

But we are not done yet. Among other features, NotePad++, when used as a code editor, highlights the language's reserved words, comments, and statements. This gives the source in the editor window a visual appeal. There is built-in support for a plethora of languages, including HTML, C, and Python, but not specifically a 6809 assembler. Fortunately, NotePad++ allows us to have a user defined language. By using this feature, we can tell NotePad++ how to highlight different aspects of our assembler source.

To accomplish this, we need a User Defined Language (UDL) definition file. Again, fortunately for us, there are people who have created such a file. The original 6809 UDL definition file was created by Steve Bamford for Michael Simonds, who was developing software for the Vectrex and later modified by Paul Fiscarelli with changes specifically for the CoCo. The UDL file can be found at the following link:

https://gist.github.com/pfiscarelli/4013e3fd743c41ffa788328fbbb4bfd9

Click on the Download ZIP and when it is downloaded extract the XML file and save it at a known location.

Having the UDL file, open NotePad++ and select **Language** from the main menu then select **User Defined Language** and lastly select **Define your language**.
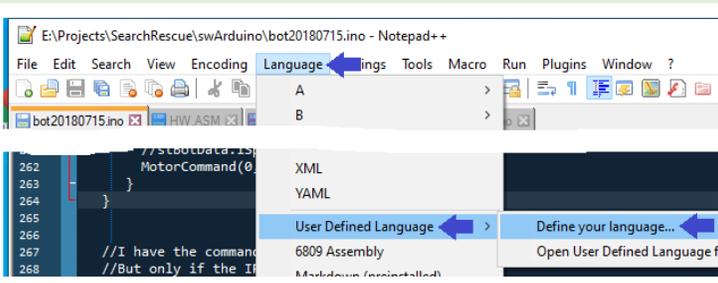
When the dialog window displays, click the **Import** button, navigate to the folder where the UDL file is located, select it, and lastly, click the **Open** button. A success indicator message will present itself. Close the dialog and you're done!
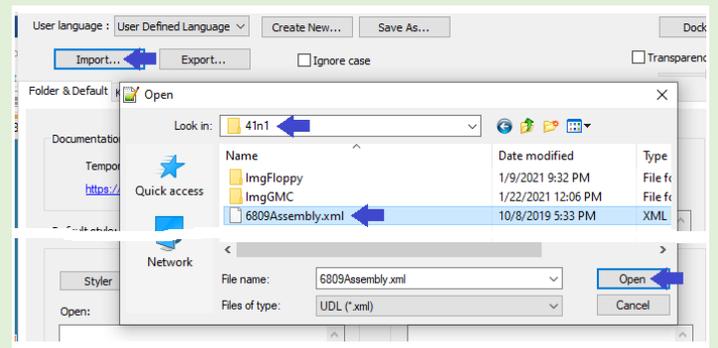
Now select **Languages** again from the main menu. **6809 Assembly** should appear at the bottom of the sub-menu.
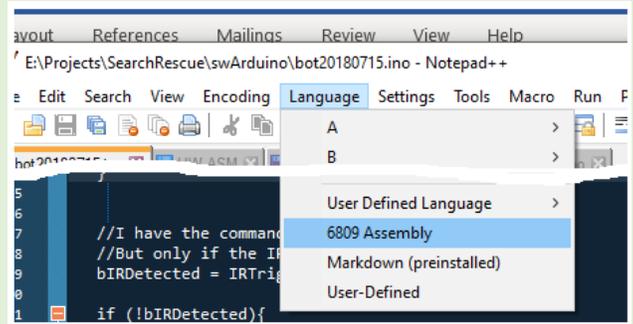
NotePad++ is now ready for 6809 assembly development!

LWTools

We now have a code editor and a CoCo environment where to test and run our program, but we are missing a key component: The assembler. This crucial component takes as input our fantastically crafted source code and transforms it to a format that the CoCo can run.

Steve uses William Astle's LWTools which is a set of cross-platform development tools for the 6809 (and 6309). Among those tools is a 6809 assembler that we'll use to convert our source to executable code. To download this software, go to the following link:

http://www.lwtools.ca/

If you have a Windows x64 installation, download the latest version which of this writing was 4.18 (Jan. 19, 2021). However, if you have Windows x86 (32-bit) then you'll need to download the previous version which is 4.17. In the main LWTools page, go to the "Third Party Contributions" section and locate the **contrib/tormod** link. Click on that to go to Tormod's download page.

Third Party Contributions

...ted here. Third party contributions are provided as is. Any problems with ...ons, I cannot provide any support for anything provided by third parties.

...g on Windows are Tormod's windows binaries in contrib/tormod. ...are provided for your convenience but are not officially supported.

Downloads

...ble. Please do not report bugs in older releases unless they are present ...

Win x32 users click on the **lwtools-4.17-win32.zip** file and Win x64 users click on the **lwtools-4.18-win64.zip** file. Note that version 4.18 only has binaries for Win x64 which is why we needed to go back one version.

Once the ZIP file is downloaded, extract the folder and place at a known location. This folder contains the LWTools.
We now have the necessary tools to develop software for the GMC.
[Newsflash: Last minute edit. Tormod Volden contacted me to let me know that he built LWTools for 32-bit Windows. Either version, 4.17 or 4.18, can be used for this article's exercises.]

Let's Test

Before moving forward, we should test our setup. To do this launch XRoar. Select **Hardware** from the main menu and then hover over **Machine**. The **Tandy CoCo 2B (NTSC, T1)** should be selected. Move the mouse pointer and hover over **Cartridge**. The **GMC Development** item should be selected. If not, there is a problem with the config file, although you can manually select the machine and cartridge for this test.
Once the correct machine and cartridge are selected, enter the following BASIC program which is based on a simple test program written by Mike Rowen in his YouTube GMC tutorial:

```
20 POKE &HFF03, 61
40 POKE &HFF23, 63
110 POKE &HFF41, 152
120 POKE &HFF41, 184
130 POKE &HFF41, 216
140 POKE &HFF41, 248
200 PRINT "PRESS ENTER KEY TO EXIT";
210 INPUT A$
310 POKE &HFF41, 159
320 POKE &HFF41, 191
330 POKE &HFF41, 223
340 POKE &HFF41, 255
```

Enter **CLS** and press **Enter**. Now enter **LIST** and press **Enter**. The screen should look like the following image.

Please verify that the BASIC program is correct. Now enter **RUN** and press the **Enter** key.
A distinct hum starts. Press any key to silence the sound and end the program. If the program behaved as described, all is working perfectly. Lines 20 and 40 enable the cartridge audio, lines 110 – 140 set the volume and lines 310 – 340 bring to volume down to silence.

Is this too much typing? Lazy… err, I mean efficient programmers can do the above a different way. Copy the program listing above into Notepad or equivalent. Save the text file to a known location. Make sure to select the **Save as type** option **All files (*.*)**. Enter the filename, preferably, but not required, in all upper case and limiting it to the 8.3 file naming convention, just to make it CoCo DOS friendly. Selecting **All files** allows us to provide an extension, which should be .BAS and not the default .TXT extension that Notepad assigns. The Notepad screen should look like the following image:

Open **File Explorer** and navigate to the folder where the text file was saved. Verify that it has the correct name. In my case, I named it GMCTEST.BAS. It should not have any other extension.

Launch XRoar and after the **OK** prompt appears select **File** from the main menu then **Load** from the sub-menu (or use the shortcut key **Ctrl-L**). The dialog that displays allows us to select the .BAS file. (Note: The **Load** option has more functionality, read all about it in the XRoar documentation.) Navigate to the folder where you saved the .BAS file and select it. Either double click the .BAS file name or click the **Open** button after selecting (highlighting) the .BAS file.

This loads the .BAS file into XRoar, but not into the BASIC environment. To complete the process, enter **CLOAD** and press the **Enter** key. After this we can **LIST** or **RUN** the BASIC program.

1. Select File/Load
2. Select .BAS file
3. Type CLOAD and press Enter key
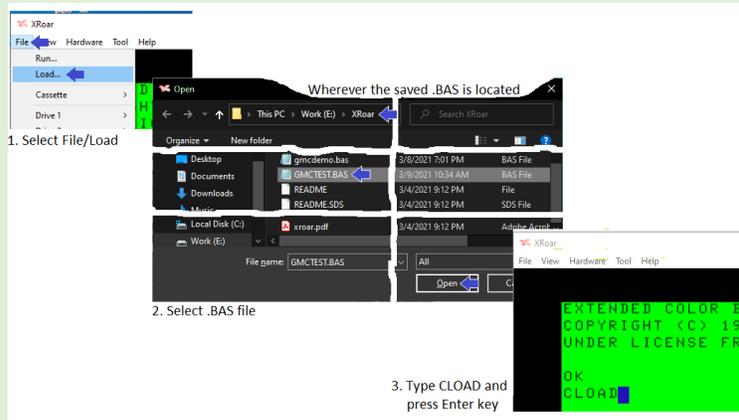
Now, we can go to John Linville's documentation on programming the GMC and go to the links that he provides for BASIC programs that demonstrate the capabilities of the GMC. Using these links, download the first BASIC program, gmcdemo.bas. After the .BAS file is downloaded, follow the technique described here to load the .BAS file into XRoar and then into the BASIC environment.

Development Environment Setup

Up till now, we've tested the GMC by using a BASIC program, but most software for cartridges is written in assembly language. This section creates a simple assembly language program that enables the cartridge audio and sets the volume. To keep it simple, that is all it does. To silence the hum, just type in some nonsense characters and press the **Enter** key to generate an **SN ERROR** which will silence the audio.

Before we enter the Twilight Zone and start coding in assembler, let's organize the tools that we have. So far, we've downloaded and set up Notepad++, LWTools and XRoar. These can be tweaked a little to make our development process better. Note that this is a matter of preference. Everyone configs their dev environment to suit their needs and tastes. This is but one example.

First, I created a folder called CoCoDev. Inside that folder I moved the LWTools and XRoar folders. I then created three new folders inside of CoCoDev: BASIC, bin and asm. We'll put our BASIC programs in the corresponding folder. The asm folder is where our assembly language source files will live,

and bin is where the machine language files generated by the assembler will be placed. I also changed the path specifications in the **Properties** dialog of the XRoar desktop shortcut. The CoCoDev folder looks like the following image now.
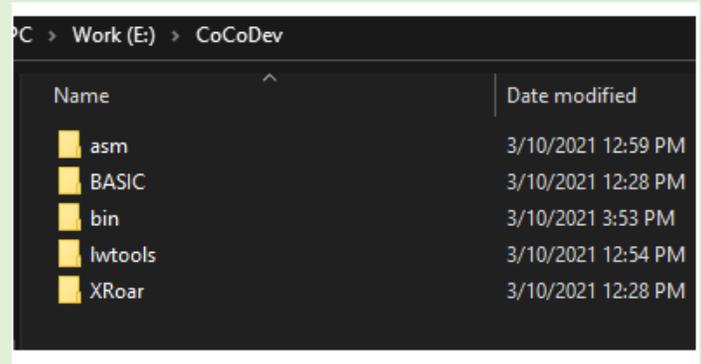
To assemble a source file, we'll need to type in a long line to launch the LWTools assembler. We need to open a Command Line window (also called a Console), navigate to the folder where the assembler is located, and then enter the command to execute the assembler program, along with the necessary parameters.

While this can be daunting, we can automate this with a venerable batch file. Copy the following script to a Notepad (or Notepad++) window and save it as asm2bin.bat in the CoCoDev folder:

```
:: Make file, by Salvador Garcia
:: Based on batch file by Paul Fiscarelli
@echo off

::Debug, remove for production
rem echo File: %1
rem echo 2nd: %2
rem pause

::Get file name into more readable-friendly variable
set filename=%~nx1
set fnamewoext=%~n1

::Set LWtools path here
set LWTOOLS_PATH=E:\CoCoDev\lwtools

::Set asm (input) and bin (output) paths
set ASM_PATH=E:\CoCoDev\asm\
set BIN_PATH=E:\CoCoDev\bin\

::Set the input file specification
set asm_fileSpec="%ASM_PATH%%filename%"

::Where to save the assembler listing
set LST_PATH="%ASM_PATH%%fnamewoext%.lst"

::No 2nd parameter then bin output, else ROM.
if "%~2"=="" goto binFile

::It is a ROM file
echo Building ROM image
set bin_fileSpec="%BIN_PATH%%fnamewoext%.ROM"
```

```
%LWTOOLS_PATH%\lwasm %asm_fileSpec% --
output=%bin_fileSpec% --raw --6809 --symbols -
l%LST_PATH%

goto all_done

:binFile
echo Building executable
set bin_fileSpec="%BIN_PATH%%fnamewoext%.BIN"
%LWTOOLS_PATH%\lwasm %asm_fileSpec% --
output=%bin_fileSpec% --decb --6809 --symbols -
l%LST_PATH%

:all_done
```

Make sure that the "%LWTOOLS_PATH%\lwasm …" line above is entered as a single line in the editor. This batch file accepts two parameters which are the file name of the assembler source and a flag to determine whether the output is a bin or rom file. It requires the file name and the extension, if any. Only the file name is required. If a full path is entered, it will be removed.

The --6809 assembler tells lwasm that this assembly language program is exclusively for the 6809. If your program uses 6309 instructions, remove this option.

The batch file will most likely require some modifications so that the paths where the LWTools, bin and asm folders are located match your installation.

To run the batch, open a command line window, navigate to CoCoDev and enter the following:

```
asm2bin myASMProgram.asm
```

Check the bin folder. There should be a myASMProgram.bin file. If not, check the batch again for any typos or bad path specifications.

Now enter the following:

```
asm2bin myASMProgram.asm we
```

If the source assembles correctly there should be a myASMProgram.rom in the bin folder. The second parameter can be any letter or number. The batch only checks whether there is a second parameter. If so, that is the flag that indicates that the output is a .ROM file.

Also check the asm folder and notice that there is a .lst file. This is a listing that the assembler creates for the source file that is being assembled. This file contains useful information. If you don't want to generate the listing, have a look at the following line in the batch file:

```
%LWTOOLS_PATH%\lwasm %asm_fileSpec% --
output=%bin_fileSpec% --decb --6809 --symbols -
l%LST_PATH%
```

Notice the last option: -l%LST_PATH%. That's a lower case "L" after the dash. If you do not want the listing, remove this option.

Now that we have the batch file, open Notepad++. Select the **Run** option from the main menu and then **Run** from the sub-

menu. This opens a dialog that allows us to define a process that will run from within Notepad++.
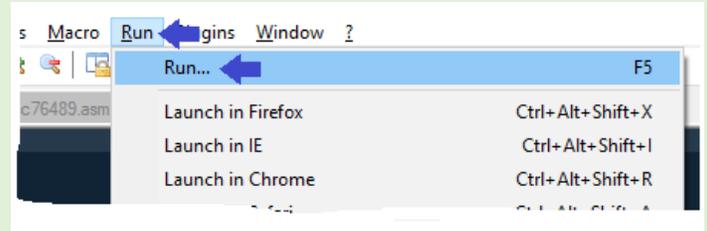
This opens the **Run** dialog. Enter the following text in the text box:

```
E:\CoCoDev\asm2bin.bat $(FULL_CURRENT_PATH)
```

If your path or your batch file name is different, make the necessary adjustments. This runs the batch file from inside Notepad++. We no longer need to manually open a command line window and run the batch file from there.
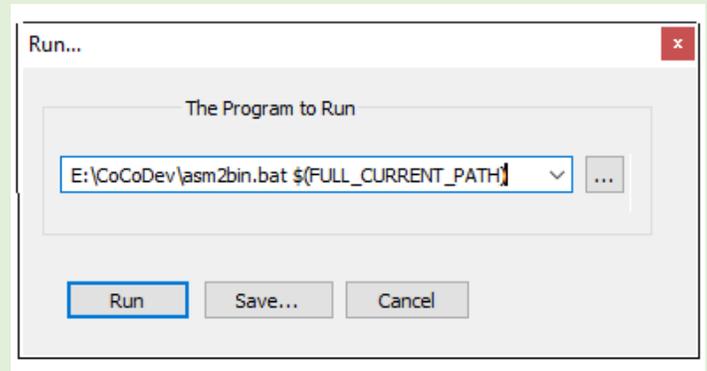

Note that the above example only provides one parameter to the batch file. This indicates that the assembler will generate a bin file. When we work on a ROM, we'll have to add a second parameter, which may be anything that isn't a blank or empty string.

Now click on the **Save** button. This opens yet another dialog which allows us to define a shortcut key. After doing research, I found that **Shift-F5** was available, so I defined that shortcut key.
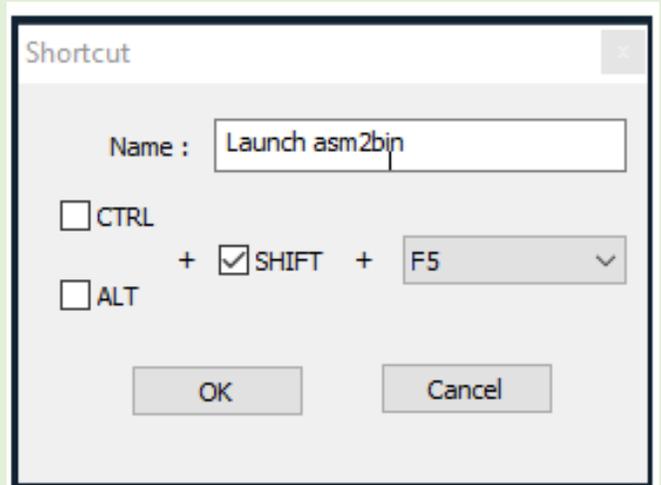
I named the shortcut "**Launch asm2bin**". Click the **OK** button to close this dialog. Also close the **Run** dialog. Again, select **Run** from the main menu. Our new command to launch asm2bin is now part of the menu!
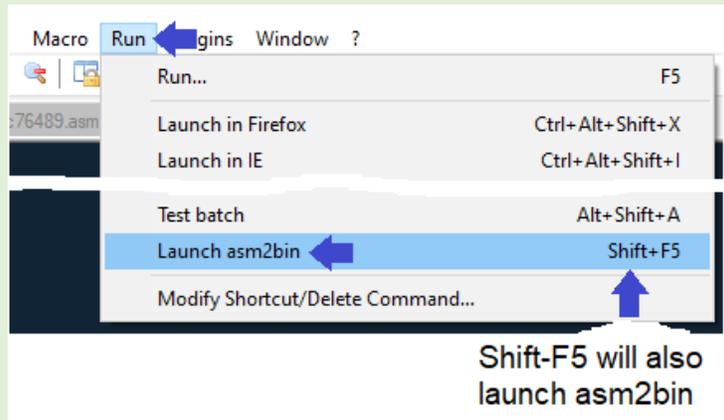
This completes the setup for the tools. Now we can get to the fun part!

GMC Cart Programming

Open Notepad++ and copy the following assembly language listing:

```
        org $C000          ;ROM code starts here
        ;org $4000           ;RAM code starts here
Start
        lda  #$34        ;Enable MUX for sound from
                            ;cartridge
        sta  $ff01
        lda  #$3f
        sta  $ff03
        lda  #$38        ;Enable sound through MUX
        sta  $ff23

        ;rts             ;Use for BASIC executables
        rti              ;Use for ROM code

        ;fill $00,(Start+$4000)-*

        End Start
```

This program enables the cartridge's sound. Since it doesn't set the volume, the default value is in effect which the maximum volume possible.

I got the lines that load the A register and then store its value in the address provided from the *Programming the GMC* documentation, authored by John Linville. At this point, the above program will run in the BASIC environment, not as a ROM cartridge. When developing complex software, I find it best to develop a module and test it in a controlled environment. Once the program works then I move it out into the wild, or in our case, to the ROM.

Save this file as "test01.asm" in the asm folder. Verify that the filename does not have the .txt extension. Now select **Run** from the main menu and **Launch asm2bin** from the submenu. Check the bin folder. There should be a file named test01.bin. That is our machine language program. This is ready to be loaded into XRoar.

Open XRoar. Verify that **Tandy CoCo 2B** is the selected machine and that **GMC Development** is the selected cartridge.

As when we used the **Load** option to attach the BASIC program to XRoar, we'll use that to load the machine language program. Select **File** from the main menu then select **Load** from the submenu. Navigate to the bin folder and select **test01.bin**. Either double click the file or click the **Open** button.

If this were a BASIC program, we would proceed with CLOAD; however, machine language files that are loaded behave slightly different. When a machine language file (.bin) is loaded, it loads into the virtual CoCo's memory, so no further LOADing is necessary. We can run the program by using an EXEC statement:

```
EXEC &H4000
```
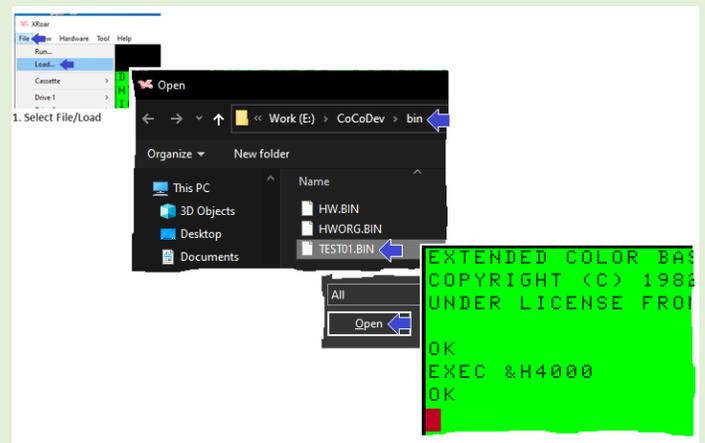

The &H4000 designation must match the address that we specified in the ORG statement in our assembler source. Once we press the **Enter** key, the humming will begin. To stop it, just enter a couple of random characters and press **Enter**. A syntax error is reported, and the sound stops.

If we heard the loud humming, then our little assembler program is correctly generating sound from the GMC. We are now ready to move this program to the ROM.

Back in Notepad++ make the following changes to the assembly source code:

```
        ORG $4000
        ;ORG $C000
```

To:

```
        ;ORG $4000
        ORG $C000
```

Change:

```
        rts             ;Use for BASIC executables
        ;rti              ;Use for ROM code
```

To:

```
        ;rts             ;Use for BASIC executables
        rti              ;Use for ROM code
```

And finally change:

```
        ;fill $00,(Start+$4000)-*
```

To:

```
        fill $00,(Start+$4000)-*
```

The first change is necessary because the ROM code is mapped to &HC000. If we place it anywhere else the CoCo will not find our code and start executing Weird Code. The second change is necessary because when we run this code in Extended BASIC, we call it as a subroutine, so we need to use the return instruction (rts), but when we run this in ROM, it is executed as an interrupt service routine, so we need to use the return from interrupt (rti) instruction. Finally, filling the binary image to 16K is not necessary for a BASIC program, but critical when run as ROM code.

Once the changes are made, save the new source. Before we can launch the asm2bin batch, we need to specify the second parameter so that the binary machine code is written to a .rom file. If we don't add the second parameter, the assembler will generate a .bin file. This will not do.

Select **Run** from the main menu and then select **Run**. The **Run** dialog opens.
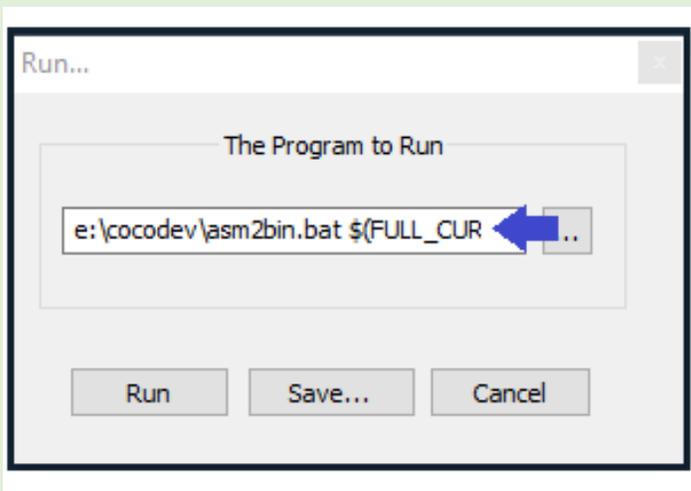
Verify that the text box contains the specification for our batch file, along with the first parameter. Select the text box and press the **End** key to move the text cursor to the end. Add a space and then any character. I chose to add "ROM" just to make it more understandable.
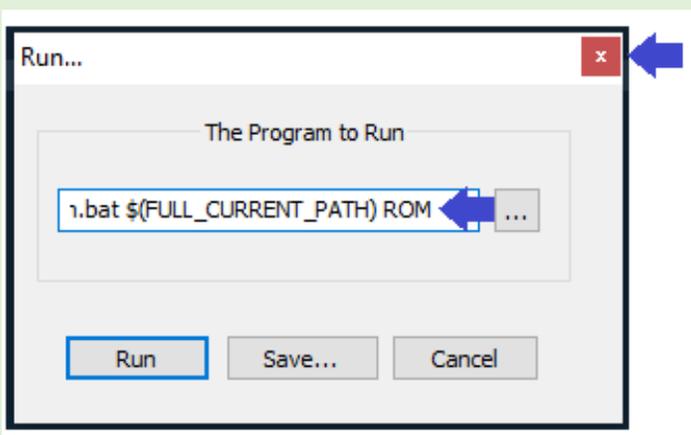
After adding the second parameter, close the dialog by clicking the **X** located in the top right corner of the dialog.

Assemble the source by selecting **Run** from the main menu and then **Launch asm2bin** from the submenu. Check the bin folder. There should be a **test01.rom** file. I had the problem that the .rom file was not generating. I solved it by deleting the **Launch asm2bin** item and recreating it with the second parameter included. To remove it, select **Run** from the main menu and then select **Modify Shortcut/Remove Command** from the submenu.

When the dialog opens, select the **Launch asm2bin** item and click the **Delete** button. Once the item is gone, close the dialog and return to the **Run/Run** dialog and recreate it with the two parameters. Test the functionality by clicking the **Run** button within the **Run** dialog. If necessary, edit the asm2bin batch file by removing the "rem" from the following lines:

```
::Debug, remove for production
rem echo File: %1
rem echo 2nd: %2
rem pause
```

Save the file then go back to Notepad++. Open the **Run** dialog, make sure that the batch file specification and parameters are in the text box and then click the **Run** button. The Command Line window should open and display the file name and the second parameter.

Press any key to continue running the batch file. The .rom file is created only if the second parameter is not empty. Don't forget to comment out the above lines in the batch file once you are finished testing/debugging.

Now we need to configure XRoar so that it will execute the ROM code. Currently it is configured so that it will not do this. Now we need to change that.

Go back to the xroar.conf file. Currently, the configuration for the GMC Development cartridge looks like the following:

```
cart GMCDev

    cart-desc "GMC Development"
    cart-type gmc
    #cart-rom MyROMSoftware.rom
    no-cart-autorun
```

We need to change the no-cart-autorun command and uncomment cart-rom, plus specify the file name of the .rom file name, including the full path. The corrected configuration should look like the following:

```
cart GMCDev
    cart-desc "GMC Development"
    cart-type gmc
    cart-rom e:/CoCoDev/bin/test01.rom
    cart-autorun
```

Of course, if your folder structure is different, the file spec for the .rom file will be different. Note that even though we are in Windows, the delimiter character used in the file path is the forward slash, not the backward slash as is customary in Windows.

We are now ready to test our small program. Verify the following:

a. The source asm file was modified accordingly and saved (very important).
b. The asm2bin batch file is correctly distinguishing between the BIN and ROM code options.
c. The .rom file is being generated in the defined bin folder (I check its time stamp to verify that it coincides with the time that I launched the asm2bin batch).
d. The XRoar config file has been modified as indicated above, the cart-rom option is pointing to the correct folder, and the config file has been saved.

Launch the XRoar emulator. Also, Steve emphasized that we don't need to quit and restart XRoar to test changes. All we need to do is do a hard reset of the virtual machine. We can do this by selecting the **Hardware** option from the main menu and then **Hard Reset** from the corresponding submenu. Or use the shortcut **Shft+Ctrl+R**.

Once XRoar launches or is restarted the hum should start. To stop it, enter any nonsense characters and press **Enter** to generate a syntax error and silence the audio.

Congratulations! You have just created your first cart program! May the future hold many more!

What's Next?

We've seen how to create a BASIC and assembler program that makes the GMC generate a sound, but this is only the tip of the iceberg. I strongly recommend that you read John Linville's documentation and watch Mike Rowen's videos. To boot, to get started on general 6809 assembly language programming, also search for Paul's videos on the Long Branch Never YouTube channel. Start at the Wiki entry for the GMC:

http://www.cocopedia.com/wiki/index.php/Game_Master_Cart

The GMC uses a complex sound generator with many virtues, but it takes time and effort to generate meaningful sounds. John has provided software to help in that aspect, along with recommendations on what to look for to learn more.

This article has provided the necessary information to set up a working development environment, giving you a push off the nest, but in the end, it is up to each one of us to learn to flap our wings so that we can soar. Good luck!

PS: Check out the complementary video:

https://youtu.be/QNR0s0_w5Tc

CoCo News!

Curated by L. Curtis Boyle
Cherry picked by:
R. Allen Murphey
Tim Lindner & Salvador Garcia

Alan from AC's 8-Bit Zone on YouTube has a video on making a 64 in 1 ROM pack, using an old game cartridge and a AM29F040 Flash chip: YouTube link


Simon Jonassen released a video on Facebook oh his updated spinning Torus demo for the Coco 1/2, showing a split screen with two different high res modes at the same time:
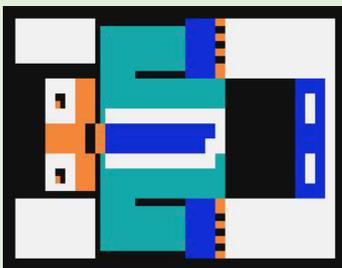

Robert Sieg put up a graphic from his MC-10 showing a Minecraft character conversion. Turns out the Minecraft characters are 32x64, and since the MC-10 (and Coco) Semigraphics-4 is 64x32, it can be converted pixel for pixel, except sideways: Facebook link

Steve Norton showed off his Atari style joystick mod that he did on his own on the Coco Facebook group: Facebook link
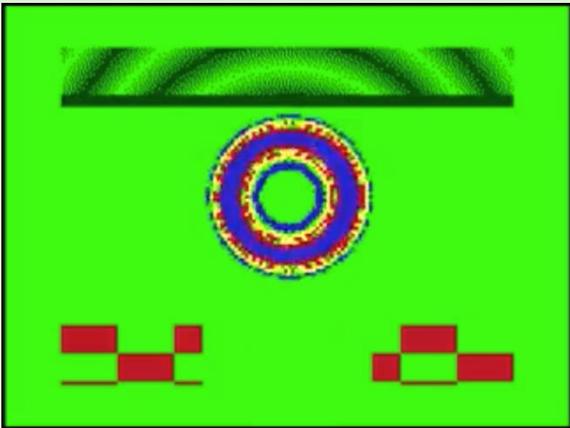
Mike Rowen of the The Coco Crew podcast has released a series of planned videos titled "Retro Printing for Fun and Profit". You can catch the on-going series on the CoCo Crew's YouTube channel.

RETRO PRINTING FOR FUN AND PROFIT
SHARING NEW & OLD PRINTERS WITH NEW & OLD COMPUTERS

Ken @ Canadian Retro Things put up a video about his TP-10 thermal printer, and also officially announced his new co-host!:   YouTube link
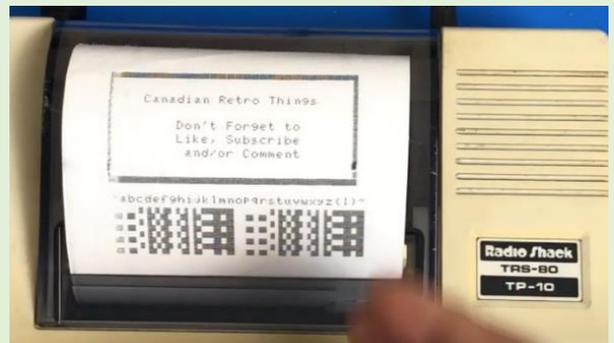
Jim McClellan has uploaded yet another update to McDraw for the MC-10 - it now allows you to erase the Alphatype (semigraphic color) letters as well as draw them, and allows the user to pick the BASIC program it generates to draw the picture (if you choose the BASIC output option). Requires the MDX32-SD. Facebook link

The Retro Channel has an update to his SVideo/Composite mod that he did back for SepTandy, and he also has a chat with Mr. Lurch about his Coco 1 and fixing up the video on that as well. He also gave a shoutout to AC's 8 Bit Zone for his video circuit as well (he also mentions that he will be doing a Component adaptor in a future episode): https://youtu.be/EonH8vyYSAs

Allen Huffman pointed to part 1 (of 8) of the BBS documentary that was done years ago and is now on YouTube. He points to a specific mention of the Coco at 21:15 in. This is an excellent documentary series on the history of BBSing (it also shows the intense rivalry between the various platforms from the time - even more intense than it is now): YouTube link
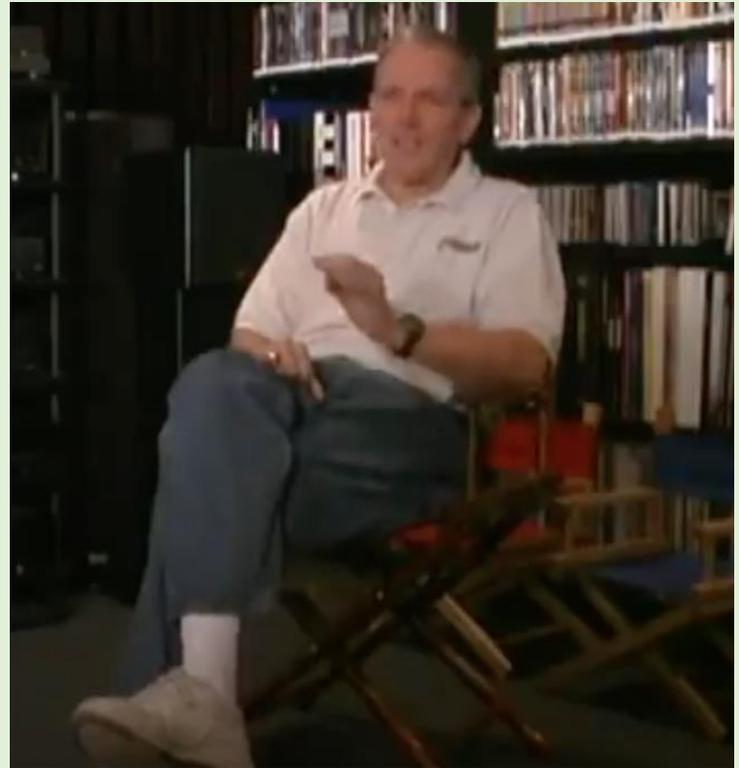
Erico Patricio Monterio linked to a video by Daniel Campos, showing a rare Coco 1 clone from Brazil called the LZ Color 64 he found at the market: Facebook link and the video itself: YouTube video

Stewart Orchard put up a video demonstrating an FPGA blitter board (that plugs into the expansion port). At this point it is more a proof of concept vs. a finished project: YouTube link

He also has the project files for it available on gitlab (where he mentions that it should also work on a Coco 1/2, and gives a plug to Jim Brain's blogs about DMA on the Coco): Gitlab link
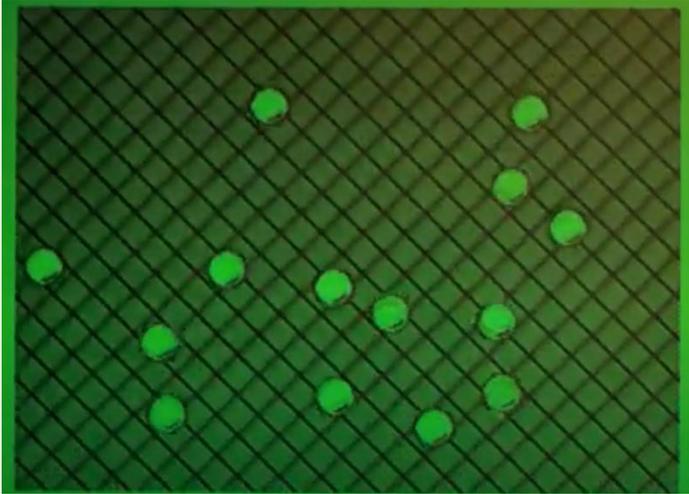
Rob's Retro Rambles has returned with a Dragon 32 game called "MacDougal's Last Stand", with his usual awesome commentary: YouTube link

Boisy Pitre has put up a new website devoted to his Ghidorah networking project for the Coco: Web/Blog link

Bill Pierce mentioned on the Coco list that VCC 2.1.0d is due out next month (April), which in addition to some bug fixes, will also have a keymap editor so that you can make custom keymaps.

There was a discussion topic brought up at the latest Glenside meeting (suggested by Boisy Pitre) about bringing about a new animal-based mascot for CocoFests. If you have any suggestions, please leave them in the comments (more details here as well): Facebook link

Erico Patricio Monteiro has a small graphical demo update showing his tweaking of his semigraphics fighting game's character graphics and a GIF of the different movements: Facebook link 1 | Facebook link 2
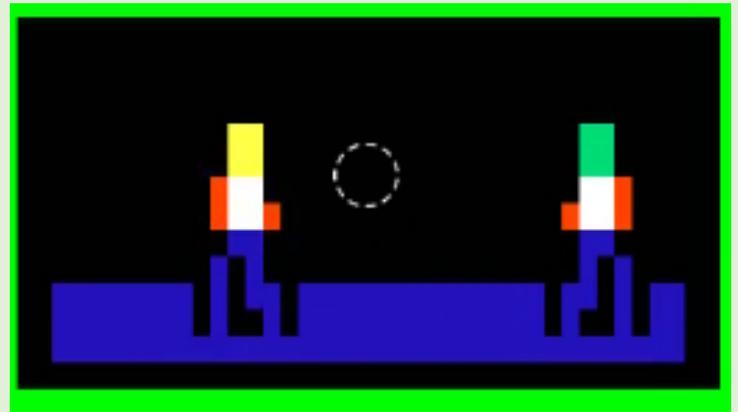
Trey Tomes put up a screenshot of a 256 width hi res screen "borrowing" a screen from Final Fantasy 2 for the Game Boy: Facebook link
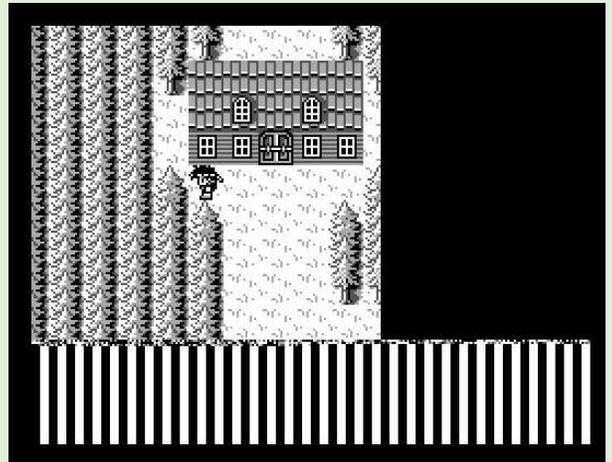

Ed Snider from the Zippster Zone has shown cool new videos on his YouTube channel, showing off his new Pick and Place machine for rapid production and 5 simultaneous displays coming out of the same CoCo 3 with the new GIME-X. Check them al out on his YouTube channel with this link.

Chad Edward has a call out to all Australian Coco users to join their Australian Coco Facebook group, in an effort to get enough people to make an Australian CocoFest/Tandy Assembly worthwhile: Facebook link

Glenside Color Computer Club's very own Brian "The Music Man" Schubring has started a YouTube channel dedicated to doing MIDI (musical instrument digital interface) on the Coco. He'll be going over how to set up MIDI and how to use Ultimuse among other things on his channel check out his YouTube channel with this link.
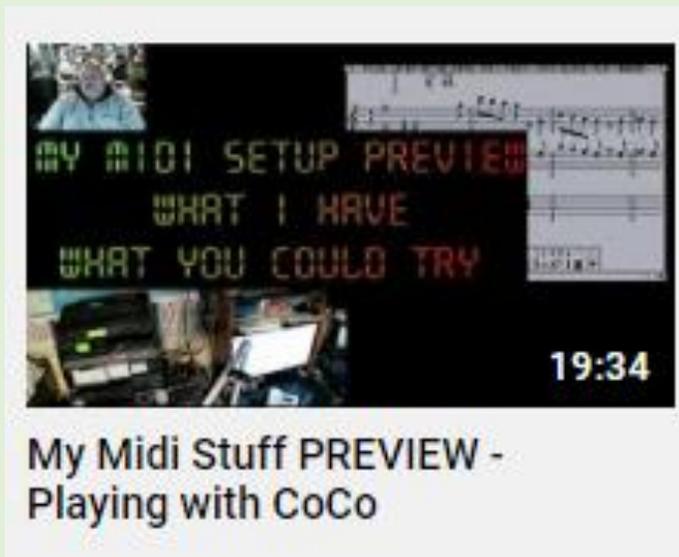

Thunder from Down Under Nick Marentes is working on a new CoCo game, "Zero Hour", follow his game development blog with this link.
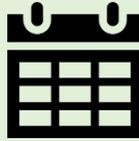

Simon Jonassen has another portion of his upcoming Coco 3 demo showing some smooth scrolling/flipping text: Facebook link

Sheldon MacDonald released a cool vector graphics (done in CocoVGA graphics modes) demo video:
Facebook link 1 | Facebook link 2 | Facebook link 3
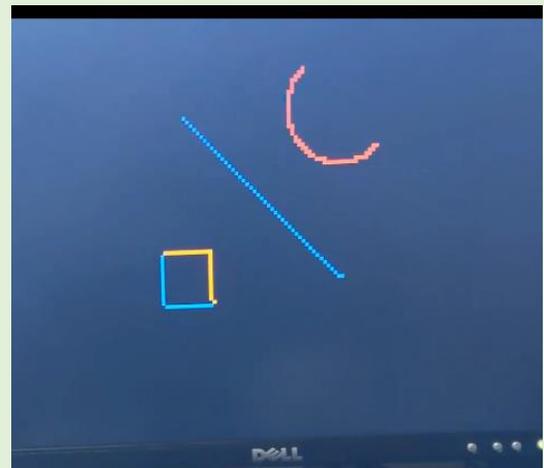
Calendar of Events

Glenside Color Computer Club, Inc. Business Meetings

Upcoming meetings: April 11ᵗʰ, May 6ᵗʰ, June 17ᵗʰ, 2021

Check the calendar on <u>our site</u> for updates.

Third Thursday of each month.
7:30 PM to 9:30 PM Central Time
Social/Project/Demos now starting 1 hour before the meeting.

BlueJeans Video Conference - https://bluejeans.com/889394355
You can call in: Phone Dial-in
+1.408.740.7256 (United States)
+1.408.317.9253 (Alternate number) (Global Numbers)
Meeting ID: 889 394 355

Glenside Virtual Spring Fest!

April 24ᵗʰ, 2021 7PM CST/8PM EST
Same BlueJeans link as the monthly meetings

A virtual social gathering of CoCo enthusiasts, no agenda, just come to mix-it-up with fellow CoCo nuts

The 29th Annual "Last" Chicago CoCoFEST!

Date: Scheduled for November 6ᵗʰ & 7ᵗʰ, 2021
Location: Holiday Inn Elk Grove, IL

http://glensideccc.com

Image courtesy of D. Bruce Moore

Tandy Assembly

October 1ˢᵗ – 3ʳᵈ 2021
http://tandyassembly.com

CoCo Community Corner

Source	Link(s)
Glenside Color Computer Club, host of CoCoFEST!	Website | Facebook
The TRS-80 Color Computer Facebook Group	Facebook
The Color Computer Mailing List	Website
The TRS-80 MC-10 Facebook Group	Facebook
The Coco Crew Podcast	Website | Facebook
The Dragon 32/64 Communities	Website | Facebook
CoCoTALK! Weekly live Color Computer talk show	Website | Facebook
The Coco Community Discord server	Get Discord | Server Link
The CoCo Show Podcast	Website | YouTube
CoCoPedia	Website
Looking for more CoCo Links? Visit I'm a CoCo Nut!	Website

(Close Parenthesis)

Thanks to D. Bruce Moore for Joey & CoCo strip #7 "Dad still upset?" and Salvador Garcia's latest entry of CoCo Chronicles.

JOEY & COCO
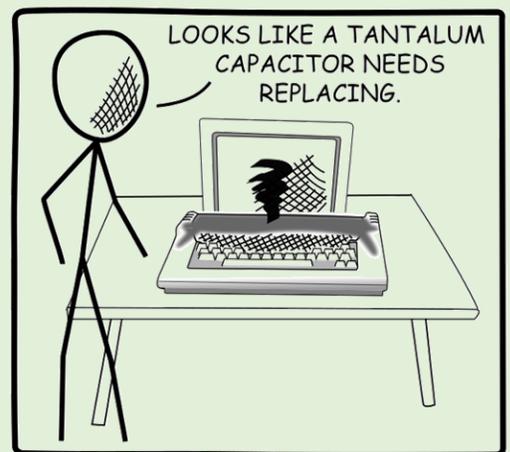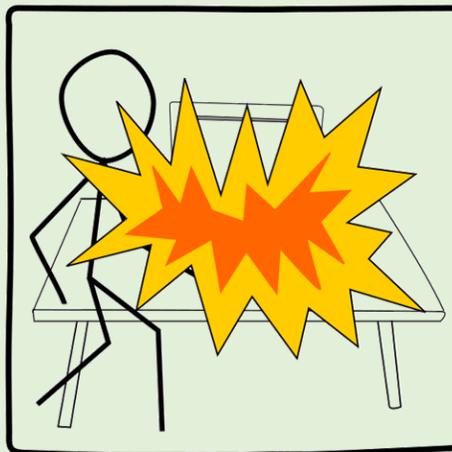
Dad still upset about recording over his ABBA tape?

Nah. And those blank Radio Shack tapes he gave me are totally tubular!

CLOAD "TICTACTO" Gunna whup you!

"Brrr-i Brrr-ee."

Bring it, Pal

What's an IO Error?

"eeee..K!"

Um...You still have that program listing?

(C) 2019 D. Bruce Moore

CoCo Chronicles — Salvador Garcia

I JUST GOT THIS COCO. FIRST TEST WITH POWER ON

©2021

LOOKS LIKE A TANTALUM CAPACITOR NEEDS REPLACING.