



Glenside Color Computer Club, Inc.
Volume 40, Number 2

Chicago, Illinois
Fall 2020

Memory Map

<u>Routines</u>	<u>Address</u>
GCCC Officers	1
CoCo~123 Information & Contributors	2
Trea\$ury Note\$, Meeting Info, Editors Clipboard	2
From the President's Platen	3
Secretary's Scroll	3
Election Results	5
President-Elect Ping	5
Thoughts on the future of Glenside	7
More on the "Change Making Benchmark"	9
Under the scope: Hires Joystick Interface	14
CoCo News	17
Calendar of Events	18
CoCo Community Corner	19
Close Parentheses	19

Upcoming Events:

Regular meetings are the 3rd Thursday of each month and are being held virtually via Blue Jeans until further notice. The next meeting is December 17th, 2020.

G.C.C.C OFFICERS

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

<u>POSITION</u>	<u>NAME</u>	<u>E-MAIL</u>	<u>PRIMARY FUNCTION</u>
President	Eric Canales	eric@canales.me	The buck stops here...
Vice-President	John Mark Mobley	johnmarkmelanie@gmail.com	Meeting planning, etc.
Treasurer	Brian Goers	briang0671@sbcglobal.net	Dues and Purchasing
Secretary	Rich Bair	mgdoc1@sbcglobal.net	Records and Reporting
Director	Tony Podraza	tonypodraza@gmail.com	CoCoFEST! Organizer
Editor	Steve Strowbridge	ogStevieStrow@gmail.com	Newsletter Production
Assistant Editor	Terry Steege	terry@terrysteege.com	Newsletter Production



Eric Canales



John Mark Mobley



Brian Goers



Rich Bair



Tony Podraza



Terry Steege



Steve Strowbridge

Copyright ©2020 Glenside Color Computer Club, Inc.
CoCo~123 A Glenside Publication Since 1985

CoCo~123 INFORMATION

CoCo~123 is the newsletter of the Glenside Color Computer Club. Your annual contribution of \$15.00 helps to keep our club going. Send your check to:

Brian Goers, Glenside Treasurer
PayPal to Brian7eg@gmail.com
Email briang0671@sbcglobal.net

Our treasury provides newsletters and good times with fellow CoCo users at our Annual "Last" Chicago CoCoFEST! and our Annual Glenside Picnic.

Should you attend the Annual CoCoFEST! your annual contribution will be covered for that year.

CoCo~123 CONTRIBUTIONS

Special thanks to our contributors this issue: Salvador Garcia, Jim Brain, Terry Steege, Rich Bair, Rick Ulland, James Jones

TREASURY NOTES



July

Beginning Balance	\$ 13,996.38
Deposits (PayPal)	\$ 14.26
Interest	\$ 0.59
July 31 st balance	\$14,011.23

August

Beginning Balance	\$14,011.23
Deposits (PayPal)	\$ 0
Interest	\$ 0.59
July 31 st balance	\$ 14,011.82

September

Beginning Balance	\$ 14,011.82
Deposits (PayPal)	\$ 45.00
Interest	\$ 0.57
July 31 st balance	\$14,057.39

Brian Goers, Treasurer
Glenside Color Computer Club

GCCC MEETINGS

The Glenside Color Computer Club meets the third Thursday of each month and are currently being held virtually through the BlueJeans video conference service.

<https://bluejeans.com/889394355>

Visit our website at:

<http://glensideccc.com>

From the Editor's Clipboard



This will be the 5th CoCo~123 that I have produced, and the role of "editor" for this project is quite the understatement. My hat goes off to John Mark Mobley and Salvador Garcia who have been the two primary driving forces in the publication for the 5 or so years I've been aware of the GCCC.

The current role of editor encompasses that of a producer, publisher, content curator, and the actual document "editing". I took on the responsibility of this role when John Mark became the VP a year ago, and after an extended period of awkward silence when a volunteer request was aired, I agreed simply because it seemed like no one else was going to.

Little did I know what I was getting myself into. While I didn't want to do this, initially, I did understand the sense of responsibility, and had an innate desire to want to preserve, if not possibly improve the look, feel, and content of the publication.

I have taken a sense of pride in producing the document and tried to bring quality and professionalism to it, and if the previous issue is any indication, I'd say the mission is starting to be accomplished.

During COVID I became increasingly busier with work and life stress, and I reached out to find a replacement, initially Terry Steege agreed to take on the role. Since that agreement, he also got busy with work, and became the vice president of GCCC (spoiler alert!)

This issue is a joint editing effort of Terry and myself, and while I would like to step down from some of the more time consuming roles of document creation and content wrangling, I'd still like to stay on board as editor, contributor and collaborator. I'd like to request a handful of people join us in creating the CoCo~123 "Dream Team" it takes a village to raise a newsletter, and I think we've got the talent to do it, so I'll be leaning heavily on our existing members of expertise such as John Mark, Salvador, Terry, myself, and I'd like to put out a call for a few more people who want to join the project and help us continue the level of excellence this publication deserves, and help us drive it into a brighter future and better versions as we continue through 2021 and beyond.

You humble editor,
Steve Strowbridge

Help us make CoCo~123 better each issue, send us submissions. Email your suggestions and contributions to glensideccc@gmail.com

From the President's Platen

Hello all. A lot has been going on around us this year. We have had to delay CoCoFEST! for the year. We have been learning to run virtual events and attend all manner of virtual gatherings in both our public and private life. Our virtual picnic on CoCoTALK turned out well and was a blast. Be sure to check it out on YouTube! Things have certainly been changing in our world with all these new virtual gatherings.

This month we have a new Glenside president, and it looks like we may have new American president. There's news of a possible vaccine for the pandemic coming down the pipe. Hopefully, the pandemic will be a thing of the past soon.

I did not get nearly as much done as I would have liked to this year due to the world seemingly stopping, but I am extremely hopeful for the future. Jim Brain, our new Glenside president, and I will continue working on the projects we started, and I think really good things are in store for the club with Jim leading the way.

CoCoFEST! 2021 is still on the roadmap for April so keep an eye on the news of the pandemic and the news coming from us. We are on the lookout for speakers this year so if you or someone you know may be interested in presenting at CoCoFEST! then please get in touch (virtual or live and in person). I hope we have a big turnout for the fest this year, so stay tuned!



Eric Canales, President
Glenside Color Computer Club

Secretary's Scroll

Rich Bair here with a summary and general commentary on the last three-monthly membership meetings, by topic:

The effort to transfer the official Glenside web site to a URL financed and managed by the club is still waiting to be finished. The ball is in Steve Bjork's court now.

The project of revision of the bylaws has made a bit of progress; there is a site up at github.com/go4retro/GCCdocs where proposed revisions can be made. A lot of questions remain to be considered.

We are still waiting for a report on our possible reporting requirements to the IRS. Brian Goers has offered to research this matter.

CoCoFEST! plans one-by-one had to be abandoned because of the pandemic. We now are looking to April 23-25, 2021; the Holiday Inn in Elk Grove Village has agreed to a contract. Now we must hope that Covid-19 will allow that plan to proceed. In the meantime, many thanks go to Steve Strowbridge for arranging a "substitute CoCoFEST!" on one of his weekly CoCoTalk programs. Our treasury is carefully guarding the money prepaid by so many of you to exhibit/attend in 2020.

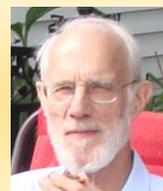
The traditional Glenside September picnic also had to go virtual. Steve S. helped us again by making the September 19th CoCoTalk double as a picnic site. You could stream video of yourself picnicking solo, while joining in on CoCo conversation.

Nominations for officers for the next year took place this month. They are:

President: Eric Canales, Jim Brain
Vice-president: John Mark Mobley, Terry Steege
Secretary: Rich Bair
Treasurer: Brian Goers

Voting is done electronically on the TandyList (www.tandylist.com/login) for all paid-up members, and it must be done prior to the beginning of the October membership meeting. To allow a bit more time for voting, we have rescheduled that meeting to a week later, on October 22nd. (Votes will not be accepted during the meeting, because it's not possible for them to be anonymous in that situation.) So even though your national vote in November is much more important, please play your part in our October election. We want our election, at least, to be seen as a legitimate expression of our members' wishes.

Happy CoCoing!.



Rich Bair, Secretary
Glenside Color Computer Club
mgdoc1@sbcglobal.net

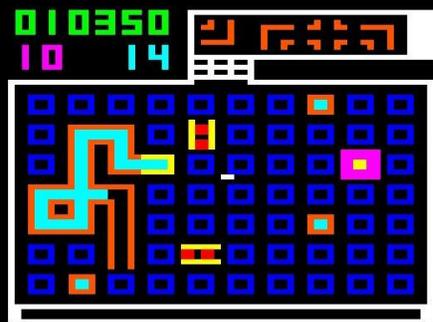
PIPES

AVAILABLE NOW!



A fast paced puzzle game in which the player lays down a set of pipes on a tiled grid in order to keep the water flowing through the pipes as long as possible without overflowing.

\$8 Digital Download
\$18 CD Media



FOR THE TANDY COLOR COMPUTER 1, 2, 3 AND DRAGON WITH 16K AND ANALOG JOYSTICK

<https://nickmarentes.com>

ELECTION RESULTS

Your 2021 G.C.C.C. Officers

POSITION	NAME		EMAIL
PRESIDENT	Jim Brain		glenside@jbrain.com
VICE PRESIDENT	Terry Steege		terry@terrysteege.com
TREASURER	Brian Goers		briang0671@sbcglobal.net
SECRETARY	Rich Bair		mgdoc1@sbcglobal.net

President-Elect Ping

As we enter into the final months of a what probably can only be charitably described as an “interesting” 2020 (was it only me that thought there would be an irritatingly significant number of “perfect eyesight” references throughout the year?), I do hope each has found time to enjoy their Color Computer (or MC-10 or Dragon) during the requisite lockdowns and “social distancing” episodes. Many I’ve spoken to throughout the year mentioned that they’ve spent more time on their setup, both as a way to “get away” from the realities of life, but also simply because they find themselves with more time on their hands, time to work on those CoCo projects that previously were indefinitely postponed in lieu of other events.

As Eric noted, this year has been a somewhat trying one for the club and its events, as well as the entire classic computer community. Physical events have given way to virtual affairs, postponements, and outright cancellations. Yet, as the name ironically suggests, the “Last” CocoFEST! moves forward. We remain hopeful that a 2021 event will happen, where we’ll plan to give the venerable Color Computer a belated but warm 40th birthday celebration. As we renew our plans for the upcoming year, please send any ideas or potential speaker requests.

As we enter the holiday season, I admit I am of an age where my childhood holiday seasons brought expectations of adding to my computer configuration. Though my parents were not wealthy by any means, and computer equipment was expensive (even more so than today) many a Christmas saw the addition of a new game, small peripheral, or subscription to a computer magazine (remember those?) where you’d devour each issue for tips and tricks. The last item proved particularly insidious, because the pages often contained advertisements for yet more cool additions to your setup.

As someone who always tries to look at the positive side of things, I hope you’ve found an opportunity in 2020 to enjoy your CoCo, whether it be revisiting a portion of your setup that you’ve not enjoyed for a while, or the purchase of a new (or new to you) item off eBay or elsewhere, and I also hope we will see each other soon in Chicago.

Jim Brain



Jim Brain, President-Elect
Glenside Color Computer
Club

COCOMAN.BIZ

Makers of Fine Fake Australian Products Since 2018

Crikey! Cable

Drivewire Cable



6ft (2m) and 10 ft (3m)
lengths available

Switch-a-roo



The One
and Only



RGB to
SCART Cable

Wallab-Y-Cable



Color Computer 3
Dual RGB Cable



Didgeridoo
Audio Digitizer Cable



Kookaburra Cable



Color Computer 3 RGB
Extension Cable

The Joey Line

Joey High Res Adapter



NEW!



Joey Serial Switch

Joey Joystick Switch



Thoughts on the Future of Glenside

I have a story to tell. It comes from a different area of my life, where I am an auto mechanic. It's about a car, but I think it has lessons that are more general.

A few years ago, my daughter bought a car, used, with close to 100,000 miles on it, but pretty nice-looking overall. There was just one problem: the "check engine" light was on. In the Chicago area we have emissions testing biennially for all cars newer than 1996, and if the light is on, that's an automatic fail. And unless you pass, you can't renew your license plate when it comes time.

Fortunately, the test isn't required immediately. They give you a year's grace. And by the time the year was up my daughter had decided to move to NYC and didn't want the car anymore. "Dad, would you like to buy my car from me?" So I did, and then it was my problem.

Again fortunately, I got a second year's grace because I'd transferred the title into my own name. But eventually I realized I had a puzzle I would need to solve before long. So the first thing I did was to buy one of these fancy electronic gizmos that communicate with the car's computer to tell you everything that's wrong with the car. (in your dreams!) The gizmo told me that it was a problem with the fuel vapor control system, and the most likely cause was a leaky gas filler cap. So I shelled out a few bucks for a new cap, used the gizmo to erase the trouble code from the computer's memory, and took a test drive. No light! But a few days later the light came on again, and I was back to square one.

The gizmo uses data collected from thousands of repair shops, so it can tell you all the components that were replaced to solve the particular trouble code you're getting. So I scanned the list, and below "gas cap" on the list was "purge control solenoid valve", so I shelled out a somewhat larger number of bucks for one of those, put it on, erased the trouble code, and test-drove. Again, no light until a few days later.

At this point I could have worked down the gizmo's list, replacing stuff until finally the light stayed off. But that might add up to a lot of \$\$ and annoyance, and there was even the possibility that it might be the gas tank itself that was leaking vapor (big \$\$ and big pain in the a**). I decided it was time to think outside the box.

There is a pressure sensor associated with the fuel tank which puts out a voltage that varies with the tank pressure. The computer uses the voltage signal to control the slight vacuum desired in the gas tank. But that voltage is not one of the signals carried to the big plug which the emissions testers use to interrogate the computer. So if I wanted to understand what was going on, I had to trace the wiring from the pressure sensor (at the back of the car) to the computer (in the front) in order to find a place where I could tap into it to read the signal myself with a voltmeter.

Soldering skills came in handy here, and soon I had a couple of wires coming up to the passenger's seat and connected to an old analog voltmeter. So I hopped in for a test drive. Glancing over at the

voltmeter every few seconds, what I discovered was this: The computer waits until the engine is fully warmed up, then puts a slight vacuum on the fuel tank. It then monitors the pressure to see if the vacuum holds. On my meter I could see the reading gradually return to the starting voltage after a few seconds. So there was a leak! The computer makes no further attempts to test and does not light the check engine light. But a "provisional" trouble code is stored. If the same code is found two more times when the car is started from cold, then the light is illuminated.

None of the above information was in the factory repair manual. But seeing the leak happening on the voltmeter gave me the confidence to resume troubleshooting.

The vapor system includes a couple dozen pipes, valves, and hoses. I bought a little hand vacuum pump, and using an approximate binary reduction process tested portions of the system separately (didn't even need the engine running to get the output from the pressure sensor) until I had isolated the trouble to an obscure valve located very inconveniently and almost out of sight at the back of the tank.

Removing the valve, I found that its white plastic body had been split by a bulge of rust forming between its body and the steel bracket holding it to the car. Replaced the valve, cured the problem. Lots of thinking, fewer \$\$.

So what does this story have to do with the Glenside Computer Club?

Our society is becoming increasingly technologically complicated, and there is a real risk of its dividing into two cohorts: a small priesthood of people who have substantial understanding of the technology, and a mass of "others" who are completely at the mercy of the priests.

I feel that "simple" computers like the CoCo give a stepping stone to understanding the incredibly more complicated ones that are in our hands (and cars) every day, whereas the leap from no understanding to a full comprehension of a modern computer is just too large otherwise. In fact, I would venture to say that there is no single person who knows and understands every detail of any modern computer down to the microcode in the processor.

So "Long Live the CoCo". May its simple innards help teach another generation of tinkerers to use their deductive abilities to solve computer problems for themselves and their friends, perhaps to eventually join the priesthood, but to always be a third cohort to mediate between the "priests" and the "others"..



Rich Bair September 2020

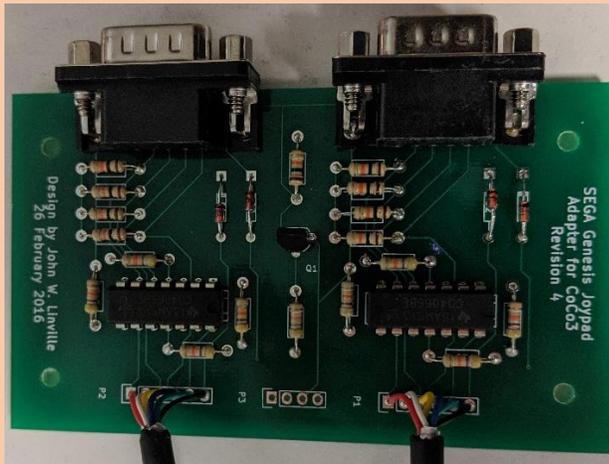
Quality CoCo Products



New Game Cartridges! \$30 each!



Flood It • Hunt the Wumpus
3D Monster Maze • Bomb Threat



CoCo Sega Genesis & Atari
Dual Gamepad Adapter \$25



New for 2020! Flood It! v1.1



The Gamester!

Pine: \$80 • Birch \$105

Custom built arcade quality controller for your CoCo!

To order any of these products, email Neil Blanchard → neil@cococrew.org

More on the Change-Making "Benchmark"

James Jones

For me, Stephen M. Pereira's article ("A Vintage Computer 'Benchmark'", *CoCo~123* v.40 #1) recalled memories of being a student assistant at the University of Oklahoma in the mid-70s.

Why? One year, an instructor teaching the usual intro to FORTRAN programming course, assigned the problem to his students, who obligingly came up with programs essentially like Mr. Pereira's "Listing 1", save that

Of course, they were in FORTRAN 66, not BASIC. They didn't take pennies in steps of five (a good optimization; the denominations of the other coins are all multiples of five, so a solution must have a multiple of five pennies--gets rid of 80% of the work). They didn't print each individual way, just the count.

As beginning FORTRAN programmers (maybe beginning programmers, too, given the era), they were constrained to a certain "job class" (it was IBM "big iron", a 370) that used WATFIV, a FORTRAN compiler from the University of Waterloo in Canada. WATFIV was meant for students. It was written for quick compilation and good error messages, *not* generating fast code. That job class also imposed a thirty-second CPU time limit, to keep new programmers' mistakes from consuming too many resources and requiring operator intervention to terminate them.

(It's hard not to laugh when typing "big iron". Hercules is an emulator for IBM 360s, 370s, and their successors. The Hercules FAQ says that a Celeron 300, ancient technology now, can emulate a 370 at speeds greater than a 3033, at least twice as fast as what OU had at the time.)

The programs all took more than thirty seconds and thus were terminated before they gave their answers... resulting in hordes of beginning FORTRAN programmers lined up to find out how to use the other FORTRAN compiler and a job class that would let their programs run to completion.

A fellow student assistant, a man I count myself extremely fortunate to have known named Raymond Schlecht (may he rest in peace), and I felt we should do something about this. I wrote a recursive program in my favorite language of the time, Algol W, and Raymond wrote a non-recursive one in FORTRAN to compile and run under WATFIV. For both, the run time was only shown in seconds, with two places after the decimal point. Mine ran in 0.01 seconds, but Raymond's took less than a hundredth of a second and thus showed as taking 0.00 seconds to run. We presented our listings and described the situation to our mutual boss, and he had a discussion with

the instructor. I like to think that the discussion was educational.

So, what's the bottleneck? Let's remind ourselves of the source. Mr. Pereira's article says the following code takes 11:10 at 0.89 MHz, 5:35 at double speed.

```
100 REM Compute how many ways to make $1.00 from
change
110 REM
115 C = 0
120 FOR P = 0 TO 100 STEP 5
130   FOR N = 0 TO 20
140     FOR D = 0 TO 10
150       FOR Q = 0 TO 4
160         IF P+(N*5)+(D*10)+(Q*25)<=100 THEN 170
162         PRINT"P=";P;"N=";N;"D=";D;"Q=";Q
164         C=C+1
170       NEXT Q
180     NEXT D
190   NEXT N
200 NEXT P
210 PRINT
220 PRINT C;"WAYS TO MAKE $1.00 FROM CHANGE"
230 PRINT
300 END
```

We can tell exactly how many times the test runs: $21 * 21 * 11 * 5 = 24,255$. (For the unfortunate freshmen, that's $101 * 21 * 11 * 5 = 116,655$.)

The first thing to note: for each of the $11 * 5 = 55$ iterations through the dime and quarter loops, $N*5$ doesn't change, but it's re-evaluated. For each of the 5 iterations through the quarter loop, $D*10$ doesn't change, but... you get the idea.

Also, must we go through all those values each time? Not at all. If P is 95, why try more than one nickel, or more than zero dimes or quarters? The values chosen earlier limit your choices for the remaining values.

It's easier to write this in terms of the values rather than the coin counts; you can figure those once you have a solution. In the compiler biz, it's called "strength reduction", and it lets us just add and subtract during the overwhelming majority of the code.

```

100 REM Compute how many ways to make $1.00 from
change
110 REM
115 C = 0
120 FOR P = 0 TO 100 STEP 5
130   FOR N = 0 TO 100-P STEP 5
140     FOR D = 0 TO 100-(P+N) STEP 10
150       FOR Q = 0 TO 100-(P+N+D) STEP 25
160         IF P+N+D+Q<>100 THEN 170
162         PRINT"P=";P;"N=";N/5;"D=";D/10;"Q=";Q/25
164         C=C+1
170       NEXT Q
180     NEXT D
190   NEXT N
200 NEXT P
210 PRINT
220 PRINT C;"WAYS TO MAKE $1.00 FROM CHANGE"
230 PRINT
300 END

```

Running that under MAME (OK, I didn't indent; I'm used to BASIC09 doing that for me) without the double speed poke took 1:13, a little over nine times faster than the original. Add a counter to see how many times line 160 is run and you'll find out that it's only doing it 1,586 times rather than 24,255.

Come to think of it, we should choose pennies *last*. The code counts on always having enough of each coin, so for any choice of a certain number of the other coins adding up to no more than a dollar, the difference is the number of pennies you need.

```

100 REM Compute how many ways to make $1.00 from
change
110 REM
115 C = 0
120 FOR N = 0 TO 100 STEP 5
140   FOR D = 0 TO 100-N STEP 10
150     FOR Q = 0 TO 100-(N+D) STEP 25
160       P = 100-(N+D+Q)
162       PRINT"P=";P;"N=";N/5;"D=";D/10;"Q=";Q/25
164       C=C+1
170     NEXT Q
180   NEXT D
190 NEXT N
210 PRINT
220 PRINT C;"WAYS TO MAKE $1.00 FROM CHANGE"
230 PRINT
300 END

```

On an emulated 0.89 MHz 6309, it runs in a bit under twenty-three seconds, over three times as fast as the previous version and nearly thirty times faster than the original. The innermost

loop necessarily runs exactly as many times as there are solutions.

(One can also do the usual dodges to reduce the enormous overhead of the Color BASIC interpreter: drop the names from NEXT statements, write the constants in hexadecimal, and cram lines together as much as possible. I leave that as an exercise for the reader.)

Now, what about BASIC09?

The BASIC09 listing in Mr. Pereira's article goes through all 101 values from 0 to 100 for pennies, putting BASIC09 at a five-fold disadvantage. Let's check...

```

PROCEDURE changeCount1
  DIM penny,nickel,dime,quarter,count:INTEGER

  count:=0
  FOR penny:=0 TO 100 STEP 5
    FOR nickel:=0 TO 20
      FOR dime:=0 TO 10
        FOR quarter:=0 TO 4
          IF penny+5*nickel+10*dime+25*quarter=100 THEN
            PRINT penny;" pennies"; nickel;" nickels";dime;"
dimes";quarter;" quarters"
            count:=count+1
          ENDIF
        NEXT quarter
      NEXT dime
    NEXT nickel
  NEXT penny
  PRINT
  PRINT count;" ways to make $1.00 from change."
  PRINT

```

(By the way, in BASIC09 you *can* use = for assignment. You have the option to use := instead. Your choice makes no difference whatsoever in code size or execution time, though I prefer := to make it clear what's assignment and what's comparison.)

On MAME emulating a 6309 with NitroS-9 Beta 5 for the 6309, running at an emulated double speed, it takes about 23 seconds, a bit over four times faster than the time listed in Mr. Pereira's article. The version with the tighter loop bounds and strength reduction? Not quite seven seconds. The pennies-last version? About four and a half seconds.

Especially now that we've cut down on needless calculation, surely we're I/O bound; time to see how long it takes just to calculate C (count in the BASIC09 version). Neither BASIC nor BASIC09 do dead code elimination, so that won't suddenly make the "benchmark" do nothing.

Using the fastest of the above versions, Color BASIC takes about six and a half seconds, so three fourths of the time it took was in the PRINT statement. BASIC09 is quick enough that my finger on the phone stopwatch app isn't good enough, so we need a wrapper to run it a lot and print time and date stamps before and after, and the result of 100 runs shows it taking between 0.21 and 0.23 seconds per run (actually a little under, because I didn't deduct for the FOR loop to get 100 runs); the version that did output was spending about 95% of its time in the PRINT statement. So when comparing just the calculation rather than calculation plus I/O, BASIC09 is about fifty times faster (taking the higher clock rate into account). Of course, it has the advantage of doing all the work with integers rather than floating point; a floating-point version takes between .59 and .61 seconds, "only" somewhere between sixteen and seventeen times faster.

(Actually, a compiler that does strength reduction probably would keep both the count and the value, which would avoid the relatively expensive division. I tried that in BASIC09 and found to my surprise that it took roughly 0.31 seconds per run. Handling those extra values all the time must be more expensive than doing three integer divides only when you actually have a solution.)

I'd have stopped here in a state of extreme smug, but one day I looked at Rosetta Code. Analogously to the Rosetta Stone, it has algorithms written in multiple languages, so programmers can be modern Champollions and learn about programming languages they don't know by comparing with those they do. They call this problem "count the coins", and they take the amount to make change for as a parameter. It turns out that if you increase the amount from a dollar, the number of ways to make change and the time that even this improved version takes grows at a serious rate.

At the time I was writing in a wonderful language, Haskell, that probably won't ever run on a CoCo, alas... but the runtimes still show the problem with the algorithm. On my desktop computer it took 0.002 seconds to solve the problem for one dollar, and 0.009 seconds for ten dollars... but then it took almost seven seconds for a hundred dollars, and I killed it after ten minutes of letting it try a thousand dollars.

If you look at these nested loops or the equivalent recursive version, you'll realize that the code is solving smaller problems over and over; the next inner loop is counting the ways to get a dollar minus what you've taken out with the other coins. Avoiding that gets us in the business of "dynamic programming" and "memoization" (thank you, Scotland and Donald Michie), spending memory to gain speed. We're also going to get out of the business of printing each way to make

change for a dollar and just calculate the number of ways it can be done.

How does it work? Think of it as adding coin denominations, one at a time, until you have them all, and at each stage figuring out how many ways there are to make each number of cents from zero to 100 using those denominations.

With zero denominations, you can generate exactly one number of cents: zero. Anything else, no way, so we have

```
PROCEDURE changeCount2
  BASE 0
  DIM i,denom(4),counts(101):INTEGER

  READ denom(0),denom(1),denom(2),denom(3)
  DATA 1,5,10,25
  counts(0):=1
  FOR i:=1 TO 100
    counts(i):=0
  NEXT i

  FOR i:=0 TO 3
    RUN add(counts, denom(i))
  NEXT i

  PRINT counts(100);" ways to make $1.00 from change."
```

leaving the question: how do you update counts to reflect a new denomination?

If $\text{count}(i) = c$, adding a denomination d means you now have c more ways to make $i+d$, $i+2d$, ... so that gives us

```
PROCEDURE add
  BASE 0
  PARAM counts(101), denom:INTEGER
  DIM i,j,c,newCounts(101):INTEGER

  FOR i:=0 TO 100
    newCounts(i):=0
  NEXT i

  FOR i:=0 TO 100
    c:=counts(i)
    IF c>0 THEN
      FOR j:=i+denom TO 100 STEP denom
        newCounts(j):=newCounts(j)+c
      NEXT j
    ENDIF
  NEXT i

  FOR i:=0 TO 100
    counts(i):=counts(i)+newCounts(i)
  NEXT i
```

Running changeCount2 gives the answer in not quite three seconds... but wait! The presence of a non-zero counts(i) only affects elements of newCounts with subscripts greater than i, so we can merge those last two loops.

```
PROCEDURE add
  BASE 0
  PARAM counts(101), denom:INTEGER
  DIM i,j,c,newCounts(101):INTEGER

  FOR i:=0 TO 100
    newCounts(i):=0
  NEXT i

  FOR i:=0 TO 100
    c:=counts(i)
    IF c>0 THEN
      FOR j:=i+denom TO 100 STEP denom
        newCounts(j):=newCounts(j)+c
      NEXT j
    ENDIF
    counts(i):=c+newCounts(i)
  NEXT i
```

That takes the run time down to not quite 2.1 seconds, still ten times longer than the nested FOR loop version.

Things to note:

For just a dollar, as we've seen, the nested FOR loop version is about a factor of ten faster. It doesn't have to fiddle with two arrays. The dynamic programming method shines when the amount increases... but the 6809's 64K address space and the limits of the numerical types mean you can't do those on the CoCo anyway.

I can't take credit for the idea. It's in the last Haskell version on Rosetta Code, which writes it as a beautiful fold that I still can't quite explain.

A Color BASIC version will be rather ugly, because Color BASIC doesn't check the first value assigned to FOR loop control variables. $i+\text{denom}$ will be greater than 100 part of the time in that inner loop in add().

One *could* speed things up at a cost in generality. Clearly after the run of add for pennies, every element of count will be 1, so just initialize it to that and don't bother to put 1 in the denom array.

Even less generally, take advantage again of the non-penny denominations and the desired sum being multiples of five, and only bother with elements of counts whose subscripts are multiples of five.

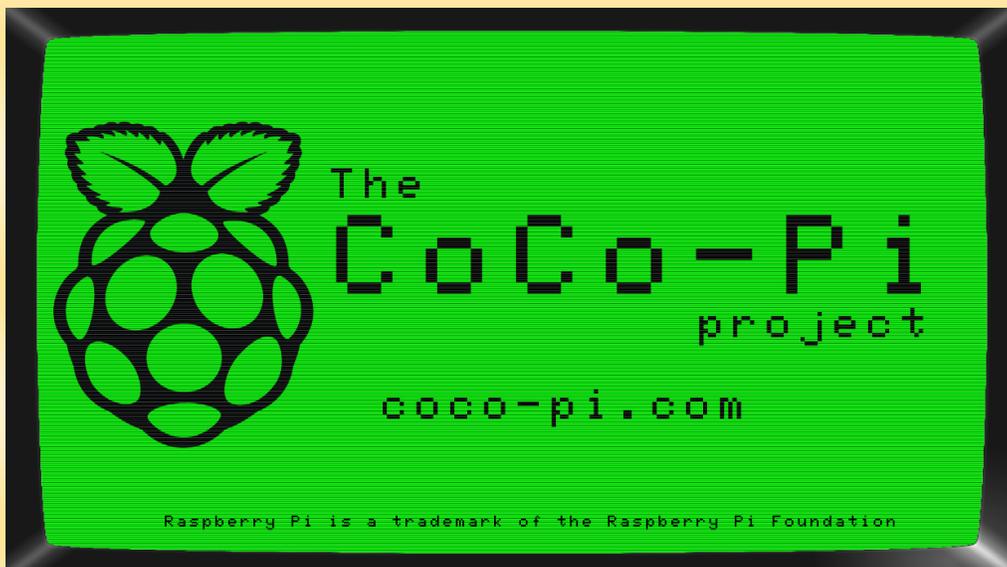
Despite all this, the point of a benchmark is to be consistent across the computers and languages tested. Unless one can run the improved versions on the machines that ran the original, there's no basis for comparison.

That said, we have learned other things as well:

Algorithms are important. A 1.79 MHz 6309 running BASIC09 I-code gets the answer in a quarter of a second, while those long-ago unfortunate engineering freshmen's brute force programs in compiled FORTRAN couldn't get results despite having 120 times longer to run.

Even an emulated 0.89 MHz 6309 running Color BASIC easily finishes before their thirty-second time limit.

An algorithm that is better for large problems may not be the way to go for small problems.





the CoCo Crew Podcast



a *delicious* adventure
into the world of
retro computing news
and information,
featuring the
Tandy Color Computer!

cococrew.org

Under the Scope: Hires joystick interface

Rick Ulland

This issue, ricku is still neck deep in joysticks so surprise! Another joystick article.

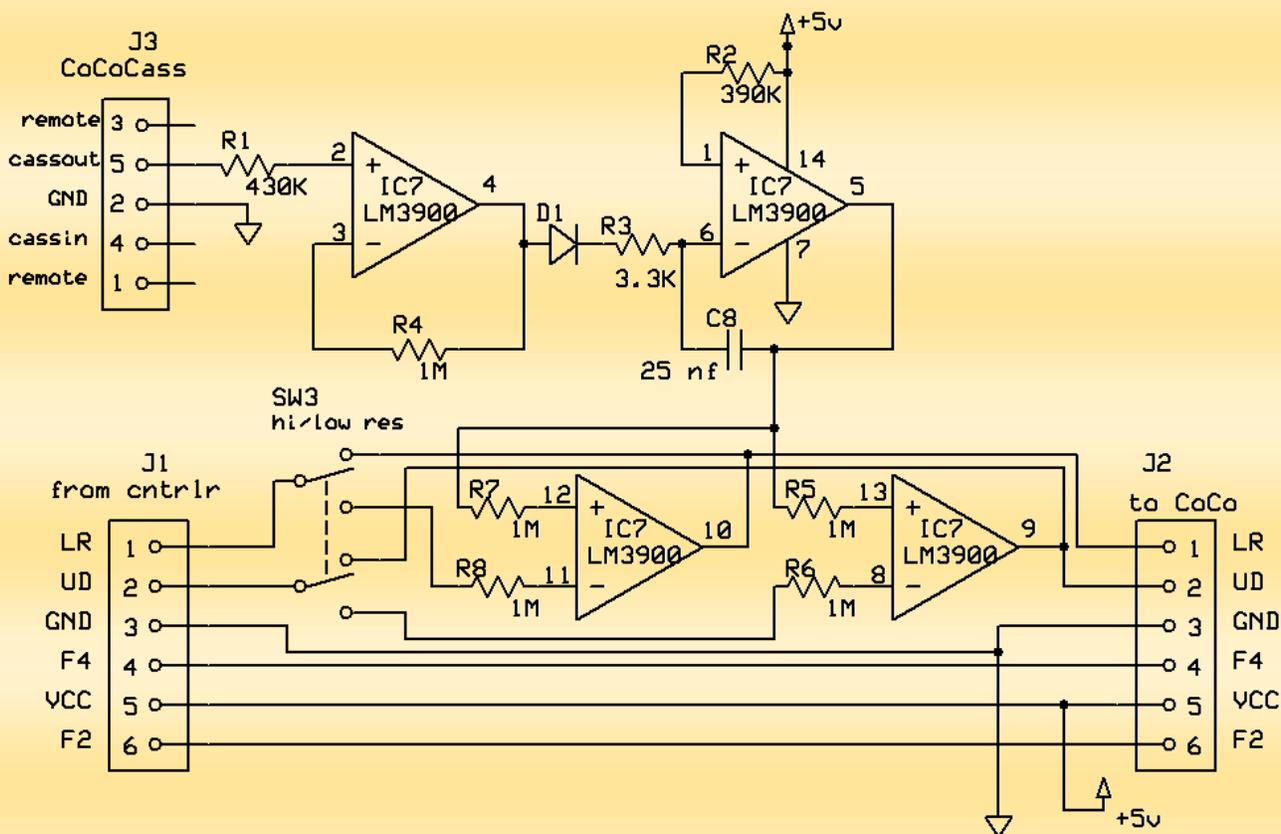
When the CoCo came out, switch based 'digital' controllers like Atari's 8 way, um, 'joystick', were common. Tandy quite rightly decided to go analog, but how analog? Bill or JR decided 64 values were enough for anybody, and so it was written - a 6 bit sample size shall be baked into the CoCo 1 hardware as well as color basic. That never changed. Even the CoCo3 redesign left the joystick alone. They weren't wrong- besides a hardware change, you'd need another SECB patch to implement, plus a backpatch to hide the new patch from any old patch or ML program. Just... no.

For those who need point to All The Things, Tandy created the 'HiRes adapter'. To sneak more data past the existing port, a cute-cute little analog computer including noise gate, function generator, and 2 pulse width modulators was cobbled up for about a Toonie (not to pander;-).

They also needed a pulse generator, but chose to have the CoCo cassette port provide that 'free'. Surely the DIN plug+handsolder cost more than a timer chip so why? OS guys? There's a Toonie init.

Anyway, if you are into hardware, it gets interesting here. The Tandy adapter appears to be a single opamp chip. These are typically voltage sensitive amplifier(s) with two inputs. + is a normal (more is more) input and - is an 'inverting' or more is less input. These are always compared, neither input by itself will do anything. Opamps have a second trick, once you push one far enough it 'saturates' and suddenly swings to max output. Analog math FTW! You don't even need to be a mathematician. Op amp cookbooks are everywhere, just look up what you need to do in general and plug in some numbers to size the R/C parts.

Except CoCo. Standard voltage oriented opamps can't quite do what we need, and the max chip count for the device was apparently one. Someone at Tandy thought about this a bit and selected a current sensitive or 'Norton' opamp. Specifically the LM3900N, four of these on one chip. That makes it worth a writeup.



Switchable Res interface

Figure 0

(Fig 0) is the Tandy circuit, or at least junkbox close. It's the one I used to get the scope traces used here. You can build this on perfboard, cut your mouse cable in half, but you'll still need that extra cassette din connector.

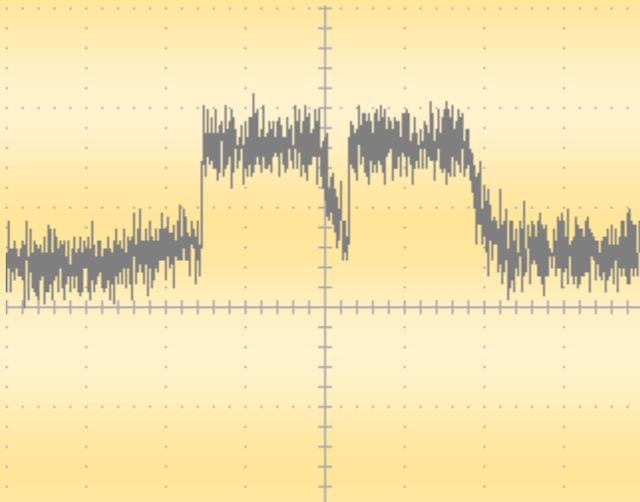


Figure 1a

If you are really into hardware, I will first apologize for not getting my scope traces all normalized, and then forge ahead with the raw analog signal from a CoCo cassette port with the hires code running. (Fig1a) is what came out of a 30 year old cable, which embarrassed me enough to solder up a new one (Fig 1b). Neither wave is exactly square, but we tried feeding both into the first opamp for giggles.

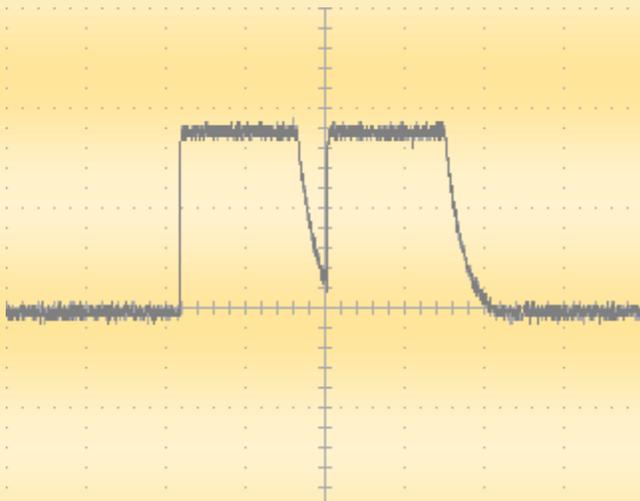


Figure 1b

This stage feeds a little of it's output back to the - input but the resistance of R4 reduces the effect until the signal pulse has really 'taken off'. Then it quickly saturates on but shuts off just as quickly on the way down. We don't care the output is inverted but do care that the noise disappears and the corners square up, mostly. Both inputs gave us (Fig 2).

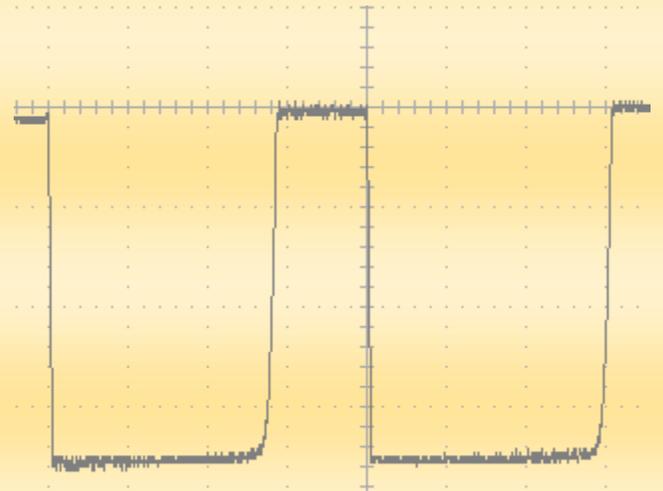


Figure 2

That cleaned up square wave is gated through a diode into a little R/C timer(ish) made by R3, C8 and the feedback from this next amp stage. The + input (thru R2) is held suitably high (compare this, buddy) and all the action is on the - side.

When a new pulse hits, C8 is empty and acts like a dead short. The amp switches full on and C8 charges. As it does, the amp reacts to the drop in charging current (not voltage) by producing a output voltage (not current) which rises in a perfect line, exactly the same way most R/C circuits don't. (Fig3). Look up 'Norton op amp integrator' on an a rainy weekend. Bring a pencil and paper, that's how we do it here:-)

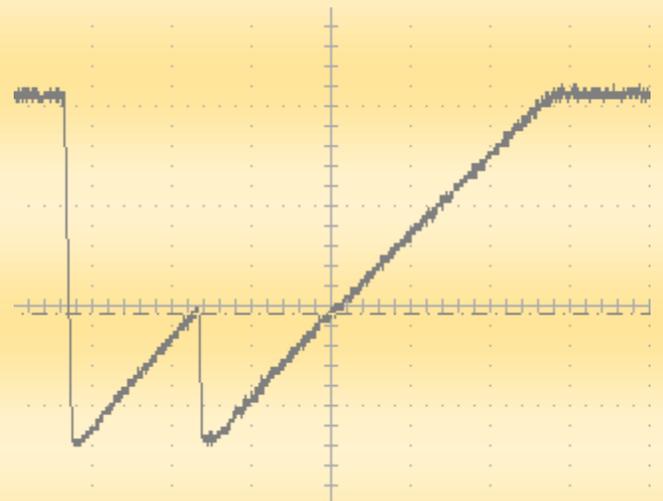


Figure 3

But wait, let's really get Norton. Feed this voltage ramp into one side of the next stage, and the output voltage from a proper (potentiometer based) joystick to the other. The LM3900 is particularly sensitive to this comparison. A little voltage difference makes a wide current swing and the amp saturates + or - close to the crossover point.

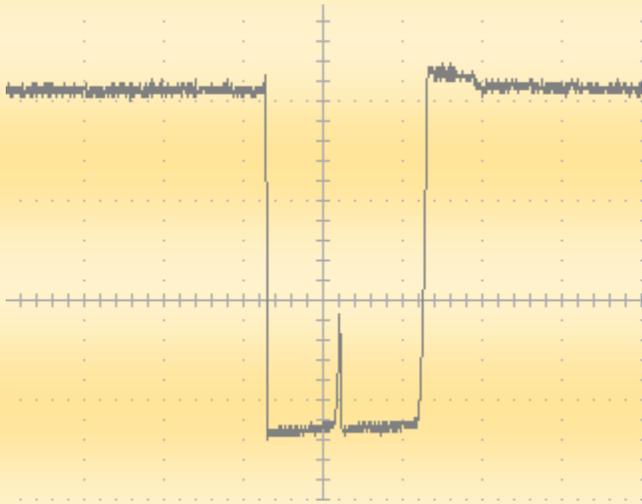


Figure 4

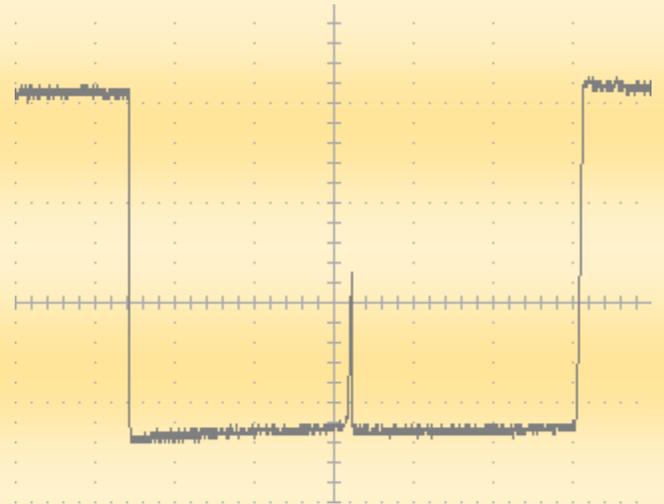


Figure 5

This makes a square wave again, from (Fig4) to (Fig5) wide, which is fed to The Ancient and Honourable 6 bit converter inside the CoCo.

This can compare nothing to infinity (OK, 63) very quickly, and this pulse length is measured somehow (OK, two Toonies) resulting in a range of values that is obviously windowed to prevent Tandy's 'balls of steel' mice from roaming across half the desktop. Your actual desktop, not the screen.

So that's the Tandy Hires. Next time, I'm sure to be into something else. See you then!

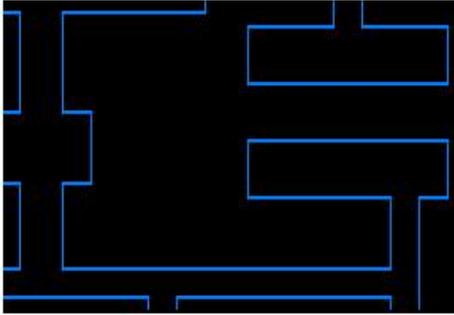
-ricku

Editor's note: I had no idea what a "toonie" was, so I asked Rick, this was his response:

"A 'loonie' is a Canadian one dollar coin which features a loon on the backside. The two dollar coin is therefore a toonie. A joke bribe to get the NitrOS9 guys to write up how the hires code works, since I haven't found a published explanation."

Temple of ROM by Rick Adams

TEMPLE OF ROM



OO



A game for the Tandy Color Computer 1, 2 or 3 (requires at least one joystick).

Temple of ROM

- The original 1984 game
- Works with Coco 1, 2 or 3

Visit: www.templeofrom.com

See rickadams.itch.io for more games!

Temple of ROM II

- New for 2020
- Still works with Coco 1, 2 or 3
- Dungeons are 45% bigger
- Two new enemies
- Build your own levels!



2020-08-01 **VCF West** happened online this year and Michael Furman shared the link to the live stream on YouTube. The live stream is now available to watch:
<https://youtu.be/7YoolSAHR5w>

2020-08-04 **Henry Rietveld** shared a link to download DOS Plus 0.3 for the Tano Dragon:
<https://www.mediafire.com/file/pj5uvjm2d2hx3pi/file>

2020-08-10 **Henry Rietveld** mentioned that he has acquired two The Olde Net WiFi modems and had tested on various retro platform and was happy with the results. The modem can be found at the following link:
<https://tinyurl.com/y57q5hue>

2020-08-10 **Ron Klein** announced the availability of MAME 0.223 for the Raspberry Pi:

Raspberry Pi 3 <https://tinyurl.com/y6ppt3l3>
Raspberry Pi 4 <https://tinyurl.com/y28lph34>

2020-08-24 **John Guin** shared a link to an article titled "Challenge to scientists: does your ten-year-old code still run?" that talks about running old code on new hardware:
<https://tinyurl.com/y69gvuwk>

2020-08-31 **Charlie** posted a link to download a disk image (Utils4OS9v1.3dsdd.dsk) containing his disk utilities for BASIC09 development:
<http://jchar.ca/coco/Download/>

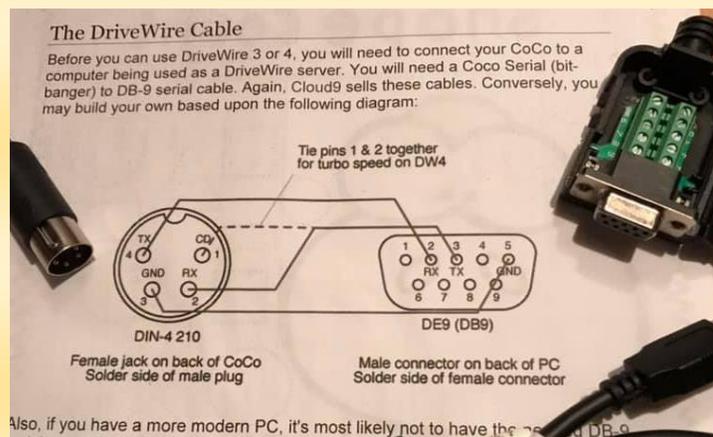
2020-09-11 **Rick Adams** announces the availability of his new game OmniStar. Come and get it for more classic gaming fun!
<https://rickadams.itch.io/omnistar>

2020-09-11 **John Mark Mobley** mentioned that the **VCF Midwest** was going to be held online on Sept. 12. Check out their YouTube channel to watch on demand, as they were recorded.

YouTube Channel: <https://www.youtube.com/vcfmidwest>
General virtual VCF Midwest page:
<http://vcfmw.org/virtual.html>

2020-09-19 **CoCoTALK!** Hosted the Glenside Color Computer Club's picnic this year! The CoCoTALK! Episode can be viewed at the following link:
<https://youtu.be/99m-DrPnKh0>

2020-09-25 **David Kroeker** shared an image of the wiring diagram for a DriveWire to CoCo cable.



2020-10-05 **Jason "CoCo Man" Reighard** announced that the Joey Hi-res joystick switch is now available for pre-orders. Get yours here: <https://tinyurl.com/y2shfb5g>



2020-10-08 **Walter Zambotti** announced that OVCC 1.4.0 with support for the Hi-res mouse/joystick is now available.

A video: <https://youtu.be/3XZWaIBRwhA>

Download:
<https://github.com/WallyZambotti/OVCC>

Calendar of Events



Compiled by Bob Swoger

Glenside Color Computer Club, Inc. Business Meetings

Future dates: December 20th2020, January 21st 2021, February 18th 2021

Check the calendar on [our site](#) for updates

Third Thursday of each month.
7:30 PM to 9:30 PM Central Time

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

The 29th Annual "Last" Chicago CoCoFEST!

Date: Scheduled for April 24th and 25th 2021
Location: Holiday Inn Elk Grove, IL

<http://glensideccc.com>

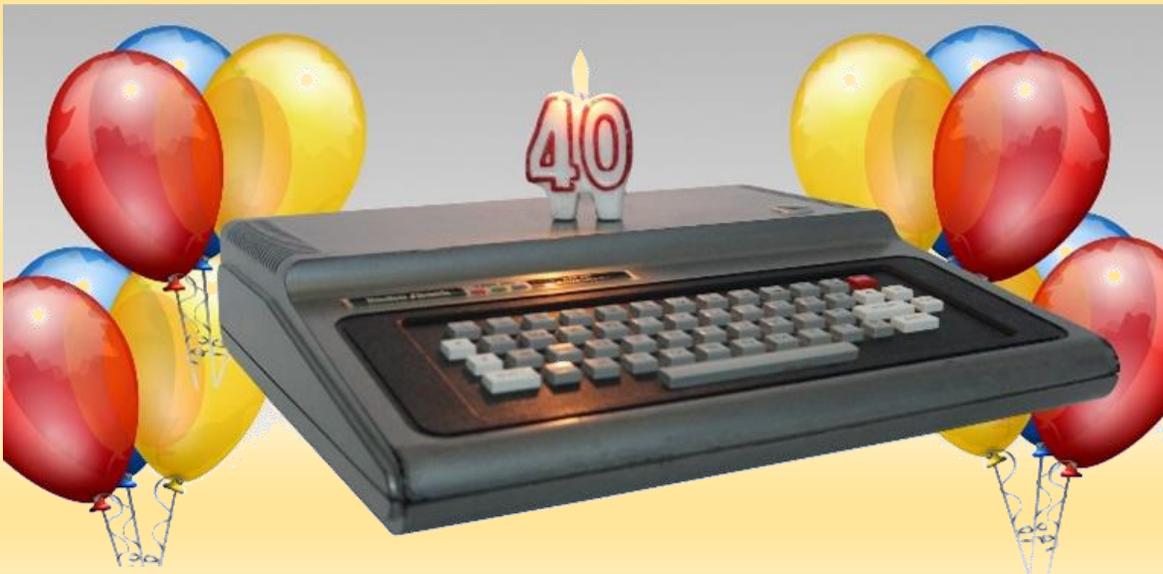


Image courtesy of D. Bruce Moore



A list of Coco web resources compiled by Steve Strowbridge

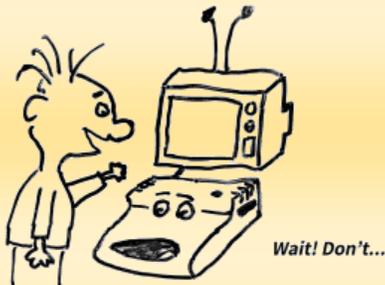
<u>Source</u>	<u>Link(s)</u>
Glenside Color Computer Club, host of CoCoFEST!	Website Classic splash page Facebook
The TRS-80 Color Computer Facebook Group	Facebook
The Color Computer Mailing List	Website
The TRS-80 MC-10 Facebook Group	Facebook
The Coco Crew Podcast	Website Facebook
The Dragon 32/64 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
Looking for more CoCo Links? Visit I'm a CoCo Nut!	Website

Close Parenthesis

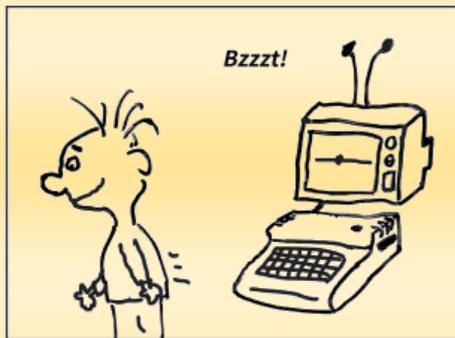
D. Bruce Moore has kindly allowed CoCo~123 to publish his new CoCo themed comic. We now bring you comic strip #5 "Wait, Don't!"

JOEY & COCO

*It was a lot of typing,
but an awesome game!
Can't wait to play again tomorrow!*

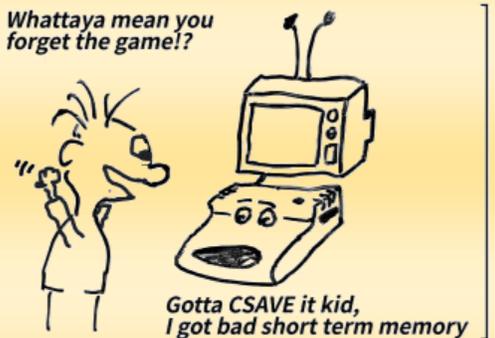


Bzzzt!



N
E
X
T
D
A
Y

*Whattaya mean you
forget the game!?*



© 2019 D. Bruce Moore

COCO TALK!



The world's leading live talk show dedicated to the Color Computer
Live and interactive every Saturday at 2PM Eastern time

Also available in audio and video podcast form

<http://cocotalk.live>