



Glenside Color Computer Club, Inc.
Volume 40, Number 1

Chicago, Illinois
Spring 2020

Memory Map

<u>Routines</u>	<u>Address</u>
GCCC Officers	1
GCCC Information, Contributors	2
Contributors to this Issue, G.C.C.C Meetings	2
Trea\$ury Note\$, Editor's Clipboard	2
3D Monster Maze review	2
From the President's Platen	3
Secretary's Scroll	3
Exploring DLOAD	5
A Vintage Computer "Benchmark"	9
New CoCo Cartridges	13
PyDriveWire DLOAD support coming soon	14
PC Jr CoCo Joystick Mod	15
CoCo News	17
Calendar of Events	19
CoCo Community Corner	20
Close Parentheses	20

Upcoming Events:

Regular meetings have been moved to the 3rd Thursday of each month and are being held virtually via Blue Jeans until further notice. The next meeting is August 20th 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	Bob Swoger	rswoger@aol.com	Newsletter Production

Copyright ©2020 Glenside Color Computer Club, Inc.



Eric Canales



John Mark Mobley



Brian Goers



Rich Bair



Tony Podraza



Steve Strowbridge

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: R. Allen Murphy, Stephen Pereira, Salvador Garcia, Christian Canales, Michael Furman, Rick Uland

TREASURY NOTES

April Treasurer Report for May 2020

April 1 st Beginning Balance	\$ 13,521.40
PayPal Transfer	\$ 534.00
Interest	\$ 0.56
April 30 th balance	\$ 14,055.40

May Treasurer Report for June 2020

Sent out two checks Sec of State \$10.00
Ronnie Qualls \$65.19 for COCO equipment.
Total \$75.19.
Deposit \$15.00,
Interest \$0.58
Totals \$15.58.

May 1 st Beginning Balance	\$ 14,055.40
Deposits/Interest	\$ 15.58
Withdrawals/Debits	\$ 75.19
May 30 th balance	\$13,995.79

June Treasurer Report for July 2020

June 1 Beginning Balance	\$13,995.79
Interest	\$ 0.59
June 30 Balance	\$13,996.38

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>

The Editor's Clipboard

We rely on folks just like you to help write articles for our newsletter. Being in COVID-19 isolation should give us all more time for the next one!

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

3D Monster Maze Review

Christian's Epic Game Reviews: 3D Monster Maze
Author: Christian Canales, roadpest58@gmail.com



3D Monster Maze by Evan Wright is a puzzle game where you try to escape a three-dimensional maze with a T-Rex that tries to eat you. This game randomizes a maze when you start, and you must figure out where the exit is by wandering down the halls. Be careful though, in some areas the t-rex will begin to follow you and might catch up to you, so you must run away.

The graphics are good, the walls are textured, and the dinosaur is detailed however the game has no sound. When I played, I was a bit confused but understood it as I went through. I even beat the game on my second try. I felt suspense from the dinosaur trying to eat me and when I finished the game, I felt relieved.

There are two different cartridge styles to choose from with their own type of box for 40\$. Overall, this game is fun to play on the CoCo and is great for any age group to learn.

Game Score/Rating: 8.5/10

From the president's platen

Hello everyone,

Sorry that this newsletter comes so late, there has been so much going on that I thought we should wait to see how conditions change in regard to Corona-virus before we publish. This last month, sadly the country has not been able to shake this virus, and so the club has decided to postpone CoCoFEST! until 2021.

We are going to postpone, rather than cancel, because we're not willing to let some virus cancel CoCoFEST! Our 29th anniversary is now scheduled for next year. We hope to have contracts signed soon.

I hope everyone is getting ready for Glenside Color Computer Club elections coming up. Voting will come faster than you think. In order to vote you must be a member in good standing. If you have paid for attending the 29th annual CoCoFEST! (now rescheduled) then there is nothing further you need to do. Your dues membership will be up to date for this year and next year. If you have not yet paid for 2020 or 2021, then I recommend getting your checks/PayPal in. Not only does it mean you get to vote in our upcoming election and vote in our general meetings, but you help to keep CoCoFEST! operating, and it helps us fulfil our mission to keep the CoCo alive.

Lastly, we are on the lookout for speakers that can come talk at CoCoFEST! in April of 2021. If we have already talked about speaking with you, then we are hoping you will still come and speak in 2021. We are looking for additional speakers, so if you have a quality presentation to share with the world, then get in touch with us!

I hope everyone has a happy and healthy 2020, and we will see you all next year!



Eric Canales, President
Glenside Color Computer Club

Secretary's Scroll

Rich Bair here with a summary of the last three membership meetings:

At the April meeting the idea of having a full-fledged CoCoFEST! later in the summer was still in play, but no venue was in place and it was realized that engaging speakers would be a problem when the date was not determined. Because of the uncertainty, no further action was taken on that matter. Attendees then focused on choosing a design for a logo/letterhead for the club. From many submissions, the one chosen shows a CoCo2 with the letters GCCC in an arc over it on a background of RGB bands, and below it an imitation of the computer's badge strip with the words Glenside Computer Club.

At the May meeting it was becoming clearer that a major CoCoFEST! in 2020 was unlikely, so we began looking forward to next year. The weekend of April 23-25 was chosen as the preferred time. This time is now confirmed with the Elk Grove Village Holiday Inn. Furthermore, they have agreed to apply our earnest money from 2020 to the new agreement, with no cancellation penalty for 2020.

By the time of the June meeting the arc of the pandemic was putting even our traditional end-of-summer picnic in doubt. As of this writing, no member has offered to host. The meeting instead focused on a few housekeeping matters:

The effort to have the club web site on a domain owned by the club is waiting on actions by Steve Bjork.

Jim Brain has assumed leadership of the group working to update the club bylaws.

Brian Goers will research what our reporting requirements to the IRS are, given that we have had an EIN from them since 2017.

Elections for club officers are coming up, with nominations at the September meeting and voting by TandyList, deadline October 15th. Give the matter some thought.

Happy CoCoing!



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

CoCo ARCADE

Dodge the red cars
and collect the flags
before fuel runs out!

For all CoCo's 1/2/3
and Dragon with
64K RAM.

RALLY-SC

Programmed by
Nickolas Marentes

TANDY
COLOR COMPUTER 1/2/3
AND DRAGON
64K RAM JOYSTICK

0000
RALLY-SC 0000
BY
NICKOLAS MARENTES
2020

0030
0025

POP STAR PILOT

Created by
Nickolas Marentes

A high flying
aerial obstacle course!

TANDY
COLOR COMPUTER 3
512K RAM
JOYSTICK

COLLECT THEM ALL!

FUEL LIVES

BONUS MULTI-FIRE TOKENS

001230

GUNSTAR
THE BATTLE FOR EARTH

TANDY
COLOR COMPUTER 3
512K RAM 6300 CPU
JOYSTICK

FOR MORE INFO GO TO: [HTTP://WWW.NICKMARENTES.COM](http://www.nickmarentes.com)

Exploring DLOAD

by R. Allen Murphey

Discord ID R. Allen Murphey#7812

June 2020 for CoCo~123

FORWARD INTO THE PAST

Extended BASIC for the Color Computer contains a command to load programs over the serial printer port.

A bare-bones command description and usage appears in the TRS-80 COLOR BASIC and EXTENDED COLOR BASIC Quick Reference [1]:
DLOAD Loads BASIC program at specified baud.
0 = 300 baud 1 = 1200 baud
DLOAD "X", 1

But how does it work and where does it load from?

In 1982, Hans Spiller at Microsoft posted "CoCo DLOAD doc" [2] from "Mark C" to the net.micro newsgroup.

This document describes in detail the protocol and structure of the communication between a Color Computer with Extended Color BASIC and a "host computer".

Everyone who bought an Extended Color BASIC-equipped computer had the client, but no details were given about the server or host that BASIC communicates with.

This changed in March 1983 when Frank Bogardus submitted the "CC DLOAD" article and source code [3] for a DLOAD "host" or "server" written in Z80 assembly language to run on the TRS-80 Model 1 computer.



CC DLOAD

by Frank Bogardus

First, read this article and learn about the Color Computer's ROM. Then you can use the DLOAD command to link it with the Model I.

```
Program Listing, Model I Host
*****
0000 *****
0001 *****
0002 *****
0003 *****
0004 *****
0005 *****
0006 *****
0007 *****
0008 *****
0009 *****
0010 *****
0011 *****
0012 *****
0013 *****
0014 *****
0015 *****
0016 *****
0017 *****
0018 *****
0019 *****
0020 *****
0021 *****
0022 *****
0023 *****
0024 *****
0025 *****
0026 *****
0027 *****
0028 *****
0029 *****
0030 *****
0031 *****
0032 *****
0033 *****
0034 *****
0035 *****
0036 *****
0037 *****
0038 *****
0039 *****
0040 *****
0041 *****
0042 *****
0043 *****
0044 *****
0045 *****
0046 *****
0047 *****
0048 *****
0049 *****
0050 *****
0051 *****
0052 *****
0053 *****
0054 *****
0055 *****
0056 *****
0057 *****
0058 *****
0059 *****
0060 *****
0061 *****
0062 *****
0063 *****
0064 *****
0065 *****
0066 *****
0067 *****
0068 *****
0069 *****
0070 *****
0071 *****
0072 *****
0073 *****
0074 *****
0075 *****
0076 *****
0077 *****
0078 *****
0079 *****
0080 *****
0081 *****
0082 *****
0083 *****
0084 *****
0085 *****
0086 *****
0087 *****
0088 *****
0089 *****
0090 *****
0091 *****
0092 *****
0093 *****
0094 *****
0095 *****
0096 *****
0097 *****
0098 *****
0099 *****
0100 *****
*****
```

Loading and saving programs via the cassette recorder limits the Color Computer to lighter computing tasks. I needed a faster way to access programs without the expense of disks and without the major program handlers needed in the Color Computer. That amounted to an impossible set of specs, until I discovered DLOAD.

The Command
Page 146 of the Extended Color Basic manual explains the command: "The statement DLOADM can be used to download (transfer) USK functions from another computer." But when I tried DLOADM as a command, I got TM error (type mismatch). Page 192 gives the syntax as DLOADM X,J. The explanation says, "... loads machine-language program at specified baud. 0=300 baud, 1=1200 baud."
Notes: this syntax also produces a

In 1983 and 1984, Spectral Associates disassembled the Color BASIC ROMs for their "Unravalled" series of books, which showed the detailed inner workings of the DLOAD client to complement Frank Bogardus excellent write-up and source code implementation of a host or server program.

The release of the Color Computer 3 with Super Extended Color BASIC 2.0 shipped without a DLOAD command implemented. The command still exists in the command table, with an assigned token, presumably for backwards compatibility loading Color BASIC programs. Any attempt to use the DLOAD command on the CoCo 3 simply triggers a BASIC warm start.

In 1988, Burt Mitchell creates a DLOAD server to run from an MS DOS PC which Bryan Clingman ported to C for use with XENIX in 1997 [4].

Fortunately for the CoCo world, Walter Zydhek recreated and distributed the Spectral Associates Unravalled [5] books again at the turn of the century, allowing CoCo newcomers and returning old-timers to continue to ponder and explore the semi-mythic DLOAD command.

With the continued rise of the Internet, Bryan Clingman's DLOAD server C code comes to the attention of Torsten Volmod who updates and cleans up the source and moves it online into a newer gitlab repository [6].

Within the last year additional interested in DLOAD appears in the form of TJBChris' DLOAD server [7], a "DSEND Command" discussion on the CoCo Mailing list [8] and TJBChris' DLOAD videos on YouTube [9].

FLASH FORWARD TO TODAY

With such a rich set of resources available, it was time to explore DLOAD for myself.

For hardware, I have a Color Computer 2B with 64K Extended Color BASIC connected via a DriveWire cable and RS232 to USB converter to a Raspberry Pi 3B+ port /dev/ttyUSB0.

For software, I run an up to date Raspbian Linux based on Debian Linux 10 "Buster".

Torsten Volmod's gitlab repository compiles into a working DLOAD server on the Pi3 with the following steps:

```
1. On the Pi, clone the repository:
>>> git clone https://gitlab.com/m6809-computer-tools/dload-server.git
Cloning into 'dload-server'...
remote: Enumerating objects: 42, done.
remote: Total 42 (delta 0), reused 0 (delta 0), pack-reused 42
Unpacking objects: 100% (42/42), done.
>>> cd dload-server
>>> make
cc      dload.c      -o dload
cc      makebin.c   -o makebin
```

Next copy a simple file into the directory as a test.

```
>>> cp ~/MC6847T1.BAS .
```

From the instructions, I first needed to run a command to setup the chosen serial port:

```
>>> sh dload.stty /dev/ttyUSB0
```

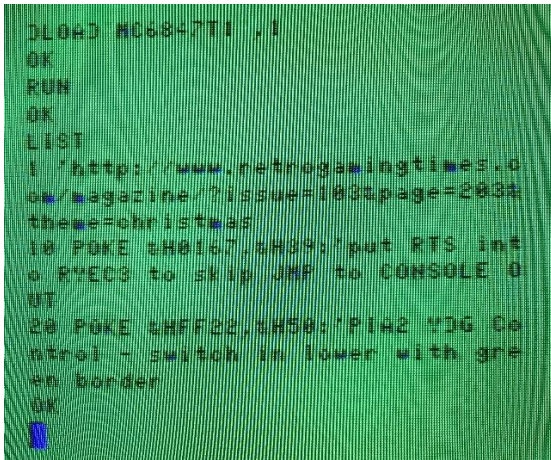
This configures the port for 1200 baud among many other settings.

Now to start up the DLOAD server and have it ready to serve an ASCII file over the serial port:

```
>>> ./dload MC6847T1.BAS a /dev/ttyUSB0
```


Finally, on the CoCo, it was time to try DLOAD:

```
OK
DLOAD "MC6847T1",1
OK
LIST
...
RUN
```



It worked the first time!

BUT WHAT ABOUT DRIVEWIRE ^[10]?

Long-time CoCoNuts are undoubtedly aware of DriveWire, a fantastic communication tool for Color Computers.

In my experimentation with DLOAD, I found it makes a nice complement to DriveWire.

My CoCo 2B does not have a CoCoSDC or HDB-DOS ROM pak to provide convenient access to a DriveWire server.

I must load HDBDOS over the cassette port from my Raspberry Pi, EXEC it, and then I can access the DriveWire server on the Pi.

That cassette load takes 57 seconds ... each time.

With DLOAD, I can skip that initial cassette load and just pull a program over serial directly, and switch to DriveWire only when I need to for larger tasks.

Access to a simple and convenient DLOAD server through the DLOAD command already present in BASIC may also lower the startup needed for newcomers to the CoCo to get more use out of their CoCo faster.

Now my Raspberry Pi can serve files to:

1. my CoCo 3 over DriveWire with pyDriveWire ^[11],
2. my CoCo 2 over cassette using mplayer <wavefile> and serial using either the DLOAD or pyDriveWire server,
3. and my MC-10 using the EmCee server in pyDrivewire.

And now yours can too.

REFERENCES

[1] Color Computer Quick Reference Manual 19xx Tandy at archive.org

https://archive.org/details/Color_Computer_Quick_Reference_Manual_19xx_Tandy/

[2] Microsoft 6809 Extended Color BASIC DLOAD/DLOADM Communications Protocol

Message-ID: <bnews.microsoft.3587>

Newsgroups: net.micro

X-Path: utzoo!decvax!microsoft!hanss

From: microsoft!hanss

Date: Sun Dec 5 14:14:35 1982

Subject: CoCo DLOAD doc. 170 lines

Posted: Sun Dec 5 00:03:33 1982

Received: Sun Dec 5 14:14:35 1982

Having seen several requests for documentation on the Radio Shack

color computer's DLOAD command, I asked the right person (Mark C,

who does not read news, nor does he have access to uucp) and got the document below in reply. Earlier I tried to get it to a previous requestor by mail and asked for acknowledgement. Since I didn't get

any I assume the paths didn't work. Submitting to news is a third and final resort.

I don't know anything about this stuff so please don't send letters about it to me. I'm just forwarding the info since I knew where it could be found.

-Hans Spiller

decvax!microsoft!hanss

Microsoft 6809 Extended Color BASIC
DLOAD/DLOADM Communications Protocol
Copyright (c) 1982 by Microsoft Corporation
Prepared by Mark L. Chamberlin
August 31, 1982

DLOAD and DLOADM send and receive packets of information to and from the host computer in order to download BASIC and machine language files. The protocol used for sending these packets was designed to facilitate detection and correction of transmission errors.

The process of downloading a file from the host involves opening a file, and then reading blocks of data from the file. The details of this protocol follow.

OPEN FILE -

1. BASIC to host - P.FILR
2. Host to BASIC - P.FILR
3. BASIC to host -
 1. 8-byte filename, left justified, blank filled
 2. XOR of the bytes in the filename
4. Host to BASIC -
 - a) If no errors detected -
 1. P.ACK
 2. file type (0=BASIC program, 2=machine language, FF=file not found)
 3. ASCII flag (0=binary file, FF=ASCII)

4. XOR of file type and ASCII flag.
- b) If errors detected, P.NAK.
5. BASIC - If errors then return to step 1.

READ BLOCK -

1. BASIC to host - P.BLKR
2. Host to BASIC - P.BLKR
3. BASIC to host -
 1. Block number (most significant 7 bits)
 2. Block number (least significant 7 bits)
 3. XOR of block number bytes

NOTE

The block number is a fourteen bit, unsigned integer in the range 0 through 16383. It is split into two seven bit values, each of which is transmitted in the least significant 7 bits of a byte.

This ensures that the most significant bit is not set except for the protocol control characters (e.g., P.BLKR). For example, a block number of 511 is transmitted as binary 00000011 and 01111111.

4. Host to BASIC -
 - a) If no errors detected -
 1. P.ACK
 2. Block length in bytes (0 through 128, 0 indicating end of file)
 3. 128 bytes of data

NOTE

128 bytes of data must be sent, regardless of the block length. If the block length is less than 128 the extra bytes are read by BASIC but not used, so, their values are of no concern.

4. XOR of block length and data bytes
- b) If errors detected, P.NAK.
5. BASIC - If errors then return to step 1.

The control character definitions are:

1. P.ACK - Acknowledge - C8 hex.
2. P.ABRT - Abort - BC hex.
3. P.BLKR - Block request - 97 hex.
4. P.FILR - File request - 8A hex.
5. P.NAK - Negative Acknowledge - DE hex.

Additional rules:

1. If errors occur during an OPEN or READ sequence, BASIC will retry the operation. For each sequence, a maximum of 5 tries is attempted. After 5 unsuccessful attempts, BASIC transmits a P.ABRT to the host and aborts the download, causing a BASIC IO error to occur.
2. If more than 10.4 seconds pass while BASIC is waiting for a byte from the host, a timeout error occurs. The operation is retried or aborted as described above.
3. If the host receives any unsolicited data, it quit transmitting and wait for BASIC to restart the

sequence.

4. The host should never time out. It should just continue to wait.

[3] 80Micro, March 1983, "CC DLOAD" by Frank Bogardus pp 190-198

<https://archive.org/details/80-microcomputing-magazine-1983-03>

[4] dload.c from dork.clingman.org via the Internet Archive

<https://web.archive.org/web/20140814190400/http://dork.clingman.org:8080/coco/>

[5] The "Unravalled II" Books by Spectral Associates and Walter Zydhek

https://archive.org/details/Color_Basic_Unravalled_II_1999_Spectral_Associates

https://archive.org/details/Extended_Basic_Unravalled_II_1999_Spectral_Associates

https://archive.org/details/Disk_Basic_Unravalled_II_1999_Spectral_Associates

https://archive.org/details/Super_Extended_Basic_Unravalled_II_1999_Spectral_Associates

[6] m6809 Computer Tools DLOAD server git repository

<https://gitlab.com/m6809-computer-tools/dload-server.git>

[7] TJBChris DLOAD server 1

https://github.com/TJBChris/dload_server1

[8] Michael Kline, Art Flexser, John Linville discuss the DLOAD on the CoCo Mailing List

[Coco] DSEND Command

<https://pairlist5.pair.net/pipermail/coco/2020-January/173443.html>

<https://pairlist5.pair.net/pipermail/coco/2020-January/173444.html>

<https://pairlist5.pair.net/pipermail/coco/2020-January/173448.html>

<https://pairlist5.pair.net/pipermail/coco/2020-January/173450.html>

<https://pairlist5.pair.net/pipermail/coco/2020-January/173452.html>

[Coco] DLOAD/DLOADM Communications Protocol

<https://pairlist5.pair.net/pipermail/coco/2020-January/173451.html>

[9] TJBChris DLOAD server videos on YouTube

Tandy Color Computer: Exploring DLOAD

<https://www.youtube.com/watch?v=-RJNheNrPB4>

CoCo 2 DLOAD Update

<https://www.youtube.com/watch?v=qboKrN3CUXI>

[10] Drivewire 4

<https://sites.google.com/site/drivewire4/>

[11] pyDriveWire

<https://github.com/n6il/pyDriveWire>

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 Serial Switch



Joey Joystick Switch

A Vintage Computer “Benchmark”

Stephen M. Pereira

On all my vintage computers, I have always enjoyed getting the same simple program to run:
How many ways are there to make \$1.00 from pennies, nickels, dimes and quarters?

Some of the computers I’ve owned came with some form of BASIC (or other language) already built in. Other computers came with no software at all, and BASIC had to be loaded into the system first, or perhaps they could run an operating system like CP/M, and some sort of BASIC could then be loaded.

The generic BASIC code for making \$1.00 from change is pretty straightforward. See Listing 1.

The program consists of 4 nested FOR-NEXT loops, counting pennies (in steps of 5), nickels, dimes and quarters. A line is printed and a counter is incremented for each combination that totals 100. Once all the loops have been performed, the total count is printed.

Over time, I’ve managed to check out a small number of systems, and I find the results interesting. This “benchmark” is likely more a measure of how efficient a particular version of BASIC is, and less a commentary on the performance of the processor, but one can see a correlation of higher CPU clock speed with faster program performance in vintage personal computers.

Here is a list of results that I’ve recorded:

System	Minutes	Seconds
Altair 8800c, 2 MHz 8080 CPU, Mini-Disk BASIC	9	4
Altair 8800c, 2 MHz 8080 CPU, CP/M, MBASIC 5.21	8	29
Altair 8800c, 2 MHz 8080 CPU, CP/M, IMSAI 8K BASIC	31	20
Altair 680, 500 KHz 6800 CPU, VTL (like Tiny BASIC)	11	13
Northstar Horizon, 4 MHz Z80 CPU, CP/M, MBASIC 5.21	4	13
HP-86, built-in HP BASIC	6	33
PDP-11/23, DEC BASIC	3	36
PDP-11/23, DEC BASICD	4	37
PDP-11/73, DEC BASIC	1	46
PDP-11/73, DEC BASICD	2	12
RetroBrew SBC, 16 MHz 68000 CPU, CP/M, BASIC	2	4
RetroBrew SBC, 16 MHz 68000 CPU, CP/M, Tiny BASIC	1	4

There have been other systems, too, like the HP-71, the TI-74, the Tandy Model 102, and the Corsham Technologies SWTPC 6800 clone, but I couldn’t find any timing results that I’d recorded at the time. I believe the all-time slowest machine was my HP-41C programmable calculator, which (if I recall correctly) took over 2 hours.

How does the Color Computer stack up against all these other vintage computers?

I currently have a CoCo 3, souped up with a 6309 CPU and 2MB RAM (Thanks to Mark Marlette and Cloud-9!), so I gave my “benchmark” a try on that:

Speed	OS	BASIC	Minutes	Seconds
.89 MHz	ROM BASIC	DECB	11	10
1.79 MHz	ROM BASIC	DECB	5	35
1.79 MHz	NitrOS9 EOU Beta 5 6809 mode	BASIC09	2	9
1.79 MHz	NitrOS9 EOU Beta 5 6309 mode	BASIC09	1	40

The results I tabulated appear to be pretty much what one would expect. In general, the faster the clock speed, the quicker the program completes. There were a few surprises for me. First, the Altair 680 with its 500 KHz clock performed far better than I might have expected. The VTL (Very Tiny Language) that came in EPROM is like the “Tiny BASICs” that were being developed by hobbyists at that time. With integer arithmetic only, it did not have the overhead required for floating point numbers, again, like Tiny BASIC.

Another surprise was the performance of the IMSAI 8K BASIC. As one of the earliest available BASIC for personal computers, my guess is that its floating-point arithmetic might have been written for precision, rather than for efficient operation. The PDP-11/23 and PDP-11/73 were expensive commercial computers at that time. One expects that they would perform this task well and they did, besting everyone else by far, except for the last surprise here. The double speed CoCo performed better than the 2 MHz Altair, and comparably well with the 4 MHz Z-80 machine.

Look at the BASIC09 performance running on NitrOS-9. That is truly impressive! The 6809-code performed right up with the best of the bunch, and the 6309-code ended up in second place overall, behind the 16 MHz 68000 running Tiny BASIC. Oh, yes, I was able to define the variables in BASIC09 as integers, so it was a fair comparison between BASIC09 and Tiny BASIC. Perhaps not so fair to compare the BASIC09 performance with the BASICs on other machines that used floating point numbers.

This program is not really a benchmark program, but it takes long enough to complete that one can see the difference between systems, and that has been interesting to me. I am sure that any number of tricks could be applied by folks looking to get the program to run faster on any given machine. As a matter of fact, while I was writing this article, I noticed that I had a discrepancy that I could not explain. My Altair 8800c at 2 MHz running MBASIC 5.21 was much faster than I expected when compared to my Horizon at 4 MHz running MBASIC 5.21 - 8 minutes 29 seconds vs. 5 minutes 22 seconds. Given the difference in clock speed, I expected that the Altair 8800c would take about twice the time that the Horizon did. I took a second look, and determined that I had used slightly different code when I tested the Horizon:

Horizon test code then:

```
160 IF P+(N*5)+(D*10)+(Q*25)=100 THEN PRINT"P=";P;"N=";N;"D=";D;"Q=";Q : C=C+1
```

Test code now:

```
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
```

Somewhere along the line, I encountered a BASIC that did not allow the all-on-one-line IF-THEN statement, and I switched the program to handle that. Running the Horizon again now with the same code that I'm using now brought its time down from 5 minutes 22 seconds to 4 minutes 13 seconds! This is much more like what I expected - just about half the 8 minutes 29 seconds taken by the Altair. The difference between checking for not equal and checking for equal saved the Horizon 1 minute 9 seconds.

When using BASIC09 on EOU Beta 5, the code had to be adjusted because BASIC09 is a structured language and will not run the generic BASIC code. I found that I had to use the := assignment instead of the regular = normally used in BASIC. As well, I had to declare all my variables in advance as integers. Finally, BASIC09 requires the ENDIF statement for any IF statement. If that is not included, then all the FOR-NEXT loops get messed up. See Listing 3.

Another favorite thing that I like to do with my vintage computer systems is to get a version of the FORTH language working on them. On most of the systems I have been able to get the original fig-FORTH to work. I also have been able to check out a few other instances of FORTH developed by others.

The FORTH code for making \$1.00 from change is also straightforward if you understand the language. See Listing 2.

Here is a list of results that I've recorded:

System	Minutes	Seconds
Altair 8800c, 2 MHz 8080 CPU, CP/M, fig-FORTH	3	45
Northstar Horizon, 4 MHz Z80 CPU, CP/M, fig-FORTH	1	53
S-100, JAIR 2 MHz 8080 SBC, CP/M, DX FORTH	1	16
Northstar Horizon, 4 MHz Z80 CPU, CP/M, DX FORTH	0	39
PDP-11/23, fig-FORTH	0	33
PDP-11/73, fig-FORTH	0	15

How does the Color Computer stack up against these other vintage computers with FORTH?

I gave my CoCo 3 and my "benchmark" a try with a couple of different instances of FORTH:

Speed	FORTH version	Minutes	Seconds
.89 MHz	Talbot Microsystems ColorFORTH	9	19
1.79 MHz	Talbot Microsystems ColorFORTH	4	37
.89 MHz	Frank Hogg Labs E-FORTH	1	8
1.79 MHz	Frank Hogg Labs E-FORTH	0	33

Again, the results are what one would expect, but there are also a couple of surprises. First of all, I was disappointed to see that the results in Talbot Microsystems ColorFORTH do not show the increased performance that I expected over DECB on the CoCo. There was an improvement, however it was much less than the more than 2X improvement seen on the Altair 8800c between BASIC and FORTH. I guess that must mean that DECB with floating point numbers has been coded to be pretty efficient.

We see that DX FORTH, available only for CP/M machines, has very impressive performance. We see the expected 2X improvement solely related to 4 MHz machines and 2 MHz machines. We see the PDP-11 systems perform impressively, as expected.

The big surprise for me was the CoCo running Frank Hogg Labs E-FORTH. At normal clock speed, this 895 KHz machine beats the 4 MHz Horizon running fig-FORTH, and rivals the 2 MHz system running DX FORTH, which was already shown to be superior. The double speed CoCo running Frank Hogg Labs E-FORTH ties the PDP-11/23 running fig-FORTH for second place and is beaten only by the PDP-11/73 running fig-FORTH!

Is there a point to me made by all of this? I think yes, perhaps a couple of points. First, it is clear, even from this unscientific study, that there is a difference between the different BASICs written by different teams that were available for personal computers during this era. BASIC was not alone in this, as we observed differences between the FORTHs that were tested here, as well. I think the big point is one that CoCo enthusiasts everywhere can enjoy: The CoCo, operating with an efficiently coded language can perform as well, or even better than other systems of the era that are perceived as much more powerful. Long live the CoCo!

Stephen M. Pereira
KB1SXE

29 May 2020



LISTING 1 - BASIC:

```
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
```


LISTING 2 - FORTH:

(How many ways to make \$1.00 from change in FORTH)

```
0 VARIABLE P
0 VARIABLE N
0 VARIABLE D
0 VARIABLE Q
0 VARIABLE C

: DOLLAR
DECIMAL
CR 0 C !
101 0 DO
  I P !
  21 0 DO
    I N !
    11 0 DO
      I D !
      5 0 DO
        I Q !
        P @ N @ 5 * D @ 10 * Q @ 25 * + + + 100 =
        IF
          ." P=" P @ . ." N=" N @ . ." D=" D @ . ." Q=" Q @ . CR
          C @ 1 + C !
        THEN
      LOOP
    LOOP
  LOOP
  5 +LOOP
CR C @ . ." WAYS TO MAKE $1.00"
CR ;
```

Listing 3 - BASIC09:

```
DIM penny:INTEGER
DIM nickel:INTEGER
DIM dime:INTEGER
DIM quarter:INTEGER
DIM count:INTEGER
count:=0
FOR penny:=0 TO 100
  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
```

New Color Computer Cartridges!

Steve Strowbridge

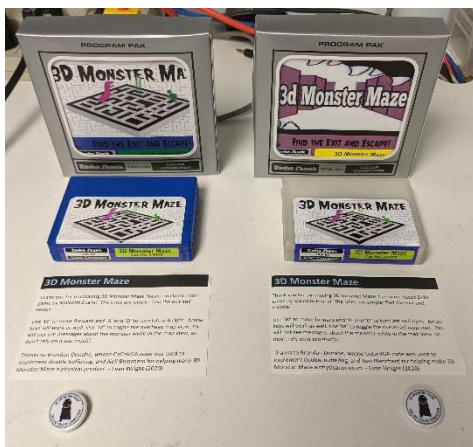
One thing I loved about my original Color Computer was the ability to slap a game cartridge into the side of it, power it on and be up and running just like any contemporary game console system of the era. As Radio Shack referred to them, “Instant loading program packs”, it was an easy way to enjoy CoCo gaming without the delay and hassle of loading a game from tape, or the extra expense of adding a disk drive system. Some of the greatest games in the CoCo library are some of the original launch titles like “Polaris” and “Project Nebula”, which still hold up as well designed and very payable games, even today.

As a gamer and collector, I was tickled to know that brand new games were being developed and produced on game cartridges. My First CoCoFEST! in 2016 gave me the opportunity to purchase several professionally produced game cartridges from John W. Linville, including Fahrfall, which were produced in brand new, shiny, injection molded plastic cartridges, including very nice boxes and labels, and a uniquely designed, custom, 3D printed

multi-colored cartridge for “Bomb Squad” by John Strong. The era of new games on cartridges was heralded in, and I have been on a mission to collect each new round of games and cartridges at all the subsequent CoCoFEST!

The 2019 CoCoFEST! brought us the Golden Edition of Fahrfall which includes a built-in music synthesizer and features multi-voice background music and sound effects, Knight’s Lore for the CoCo 3, and a cool, Hunt the Wumpus game from Evan Wright.

Sadly, we have not had an in-person CoCoFEST! in 2020 yet, however, that has not stopped Neil Blanchard of the CoCo Crew Podcast from bringing new cartridges to the CoCo community this year, including several new color choices! Unlike some of the previous professionally molded cartridges produced that came in your choice of red, Neil has custom 3D printed CoCo cartridge cases in a few colors, including a very cool looking translucent one.



The new cartridges this year include 3D Monster Maze, from Evan Wright, this being now his 3rd CoCo Cartridge, and the re-release of Rick Adams’ Bomb Threat on a new, custom yellow cartridge which looks fantastic with the artwork and label. 3D Monster Maze can be obtained in either a blue or translucent cartridge. They are packed with an insert card that includes instructions, in the case of 3D monster maze a cool button, and are packaged in the new, high quality program pack reproduction boxes produced by Tim Linder,

which are of a higher quality stock than the original, and look fantastic. They are then sealed in a plastic bag that can be opened and resealed as needed.

The price for these new cartridges is \$30 each, and they can be obtained by sending an email to Neil Blanchard via Neil@cococrew.org to arrange the shipment. Thank you, Neil, for bringing us new games to play and collect on our CoCo.

pyDriveWire DLOAD Support is Coming Soon

By Michael Furman

There has been a lot of discussion lately about the DLOAD protocol. The recent resurgence started with TJBChris's excellent YouTube Videos about the DLOAD Server he wrote which can run on Zenix on his TRS-80 Model 16 server.

Elsewhere in this issue R. Allen Murphey talks about running a DLOAD server which has been available for a long time. As some of you know I am always looking for feature requests for pyDriveWire. Allen had asked me if it was possible to add DLOAD support to pyDriveWire, and my response was "Yep, I think that's possible".

A few weeks later I found enough free time to get it done and pyDriveWire's DLOAD support was born! Testing is in progress and I currently plan to have it ready for the next minor release of pyDriveWire v0.5d which should be published near the end of July or early August 2020.

R. Allen Murphey's article provides a description of the protocol. Allen sent me a copy of this article and while studying it I found that the DriveWire and DLOAD protocols use the same bytes for some of their commands. A fancy way to say this is that the two protocols are not "orthogonal". This means that unlike the EmCee protocol DLOAD support will need to be enabled when needed and disabled when you want to go back to EmCee or DriveWire.

This can be done through new commands being added to pyDriveWire's Command Set. Issuing the command:

```
pyDriveWire> dload enable 1200
```

Will switch on the DLOAD protocol at 1200 baud. You could put any baud rate you want there, but the CoCo only supports 300 (,0) and 1200 baud (,1)

Next you need to tell the pyDriveWire server where to look for the files the CoCo is asking for. The command for this is:

```
pyDriveWire> dload setdir /path/to/dload/files
```

Now if you placed a file named "FOO" in the /path/to/dload/files directory, you could either do:

```
CoCo: DLOAD "FOO",1  
or  
CoCo: DLOADM "FOO",1
```

to load the file in.

One drawback of the DLOAD protocol is that it only supports an 8-byte long file name. This pretty much means that file name extensions go out the window. While you could do:

```
mv FOO.BAS FOO  
or  
mv FOO.BIN FOO
```

Even though the two files are really different files you can only have one of them online at a time. pyDriveWire provides a way to manage this called "aliases". You could set up aliases like this:

```
pyDriveWire> dload alias add FOOBAS FOO.BAS  
pyDriveWire> dload alias add FOOBIN FOO.BIN
```

Now from the CoCo side you can load both files:

```
CoCo: DLOAD "FOOBAS",1  
CoCo: DLOADM "FOOBIN",1
```

This is also handy for something that you load frequently such as HDBDOS. You could create an alias for it and put it in your pydrivewirer configuration file so it's always there. What follows is a procedure to BootStrap HDBDOS on a CoCo2.

```
pyDriveWire> dload alias add HDBDOS hdbdw3cc2.bin  
pyDriveWire> dload enable 1200
```

```
CoCo: DLOADM "HDBDOS",1
```

```
pyDriveWire> dload disable
```

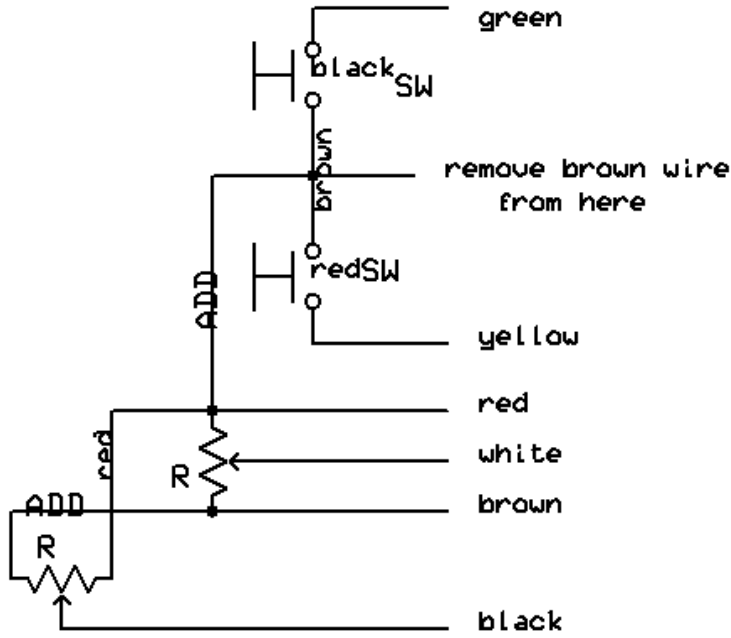
```
CoCo: EXEC
```

Now your Coco is running HDBDOS and since DLOAD is disabled DriveWire is enabled again!

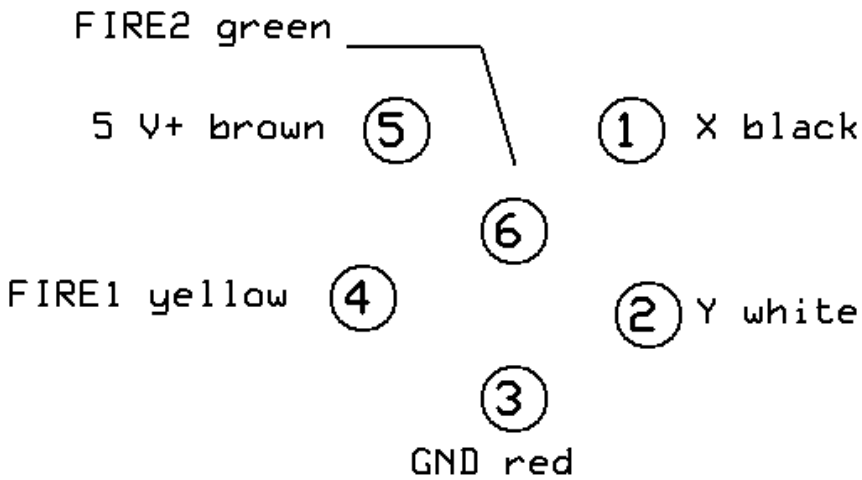
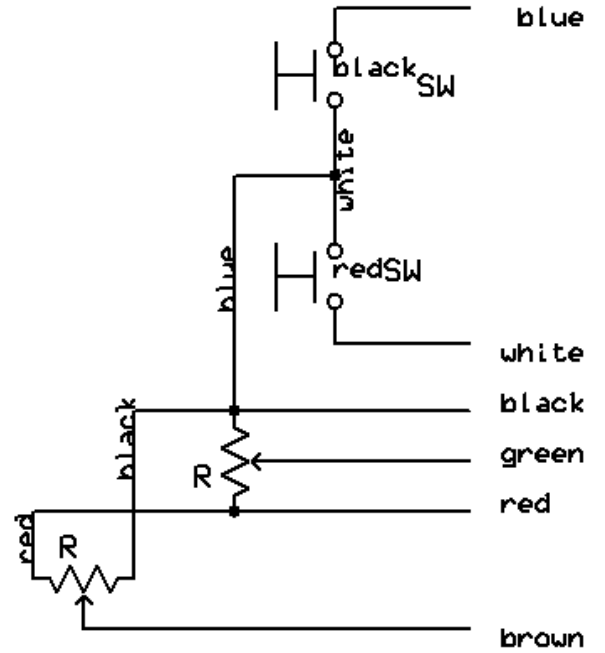
PC Jr -to- CoCo Joystick mod

By Rick Uland

PC Jr Joystick



CoCo3 Joystick



PC colors to CoCo connector
from wire side of connector

PCjr vs CoCo (Variable resistors vs. potentiometers)

For whatever reason, brand new 'IBM PCjr' joysticks have been hitting the surplus market. These are physically the same as the Tandy deluxe joystick, and if you can solder, the electrical issues are easily fixed. All you need is the proper DIN6 connector or the cable from a broken Tandy joystick, and 4 inches of jumper wire. Wanna hear it? Here it goes.

Get the stick in spring-centered mode before opening and remember, cell phones have cameras. Take a picture before you change things or having things change for you.

IBM has no steenkin' warranty sticker, remove 2 screws, then grab the joystick stick while removing bottom case half. For here on, remember the joystick assembly is 'slide in place'- leave it on it's side so you dont push things apart by the handle, maybe a instagram pic while it's together.

The diagram works like this. If you remove the case bottom, with the cable coming in on top, the physical connections are as drawn. The wire colours down the right side are the standard PCjr and CoCo input cable. Along internal jumpers are std? colours or ADD when a jumper needs to be added. If your joystick agrees, this will work...

----- A new joystick and a new connector for the CoCo end of the cable.

We are using the IBM cable and wire colours, ignore the CoCo3 joystick portion of the chart.

The input cable has a brown wire that leads to both fire buttons. Cut this wire near the fire buttons, leaving the buttons connected together.

Connect the newly freed brown wire to the unused pin on either potentiometer, then a jumper wire to the unused pin on the other.

Locate the red wire (other side of both pots from our new jumper). Run a new jumper from there to the recently abandoned brown wire connecting both fire buttons. Finally, whack off that weird PCjr connector and add a regular DIN6. PUT THE PLASTIC SHIELD ON NOW. Did you put the plastic shield on? No, really. Is it on the cable?

Connect the PC coloured wires as shown - this chart is 'as you solder' from the wire side. Done!

Note: There be capacitors. They be fine. See below.

----- A new Joystick and a CoCo cable

To reuse a whole CoCo cable, compare the left and right charts- it's a simple color swap plus add the two jumper wires.

And so why have we done this?

There are two ways to make an analog measurement. Both use the same tool, a strip of resistor with a wiper that runs along it.

The PCjr method is to put a voltage on one end of that strip, leave the other free, and read what comes back from the wiper. This is a variable resistor. You measure it by pulling current through this resistor and checking the voltage drop.

The CoCo method is to put voltage on one end of the strip, ground on the other end, and sniff the voltage returned on the wiper. This is a potentiometer. You try to measure this voltage 'in a vacuum' at no current.

So, why?

The 'variable resistor' design feeds a known voltage out the cable, into the control's resistor strip, across the wiper connection, and back down the cable. The control voltage is formed and measured at the same place - the end of a 12 foot path of skinny wire and cheap connectors.

The 'Pot' design delivers ground and VCC to opposite sides of the control's resistor strip. The control voltage is formed along that resistor. So, for example, there is still a wiper connection to pick off the relative voltage, but we are not pulling current through that wiper to create the voltage.

Normally, 'variable resistor' joysticks would be a wire short, but IBM decided the variable resistors would use 5v and the fire buttons would use ground. So the new cable has the 'extra' wire we need! Thanks, blue.



CoCo News

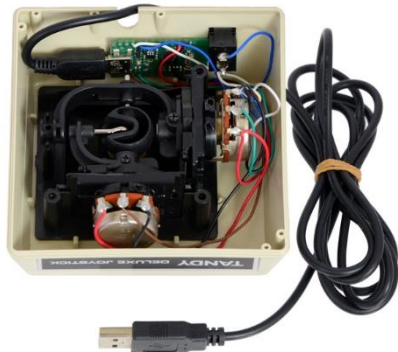
Compiled by
Salvador Garcia,
Steve Strowbridge

Michael Furman mentioned a C Cross compiler. The Github repository is here:

<https://github.com/n6il/MicrowareCCC>

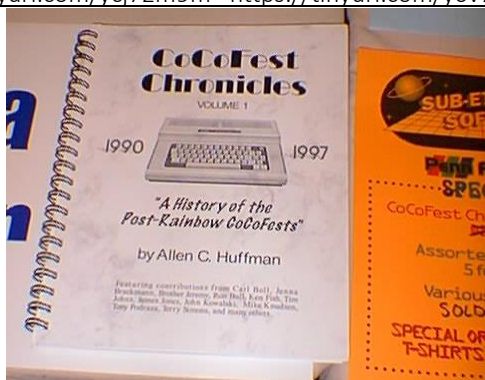
Henry Rietveld posted a link to a Pololu article that presents a process to convert a Tandy Deluxe Joystick to USB.

<https://www.pololu.com/docs/0J59/all>



Allen Huffman announced that his book, "CoCoFEST! Chronicles" is available for download as a PDF file and available to order a print.

<https://tinyurl.com/yjc72m9m> <https://tinyurl.com/y6v7lsbl>



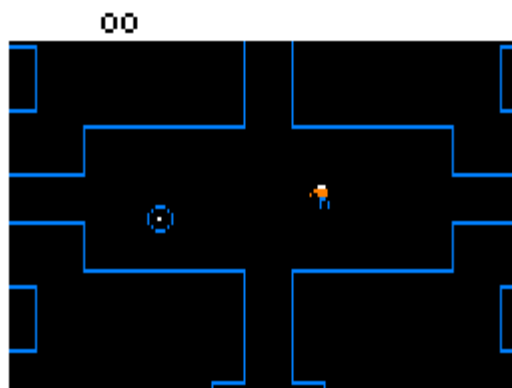
Ed Snider, aka, Zippster, announced that the beta release of the GIMEX was ready on April 23, 2020. He was looking for volunteers to help test the product before final release. All beta slots were filled quickly. Project page for the GIMEX follows:

<https://tinyurl.com/yb6deeh1>



Rick Adams has now released Temple of ROM II. The game is downloadable as a disk image and is free of charge, although a donate button is available for those who wish to shower Rick with riches.

<http://templeofrom.com/>



Tormod Volden packaged his CMOC compiler v0.1.65 so that it can run on Windows without the Cygwin environment. Install instructions are: Download package, unzip file, and add the cmoc-win32 folder to the PATH variable.

<http://toolshed.sourceforge.net/cmoc/>

Brett Gordon released version .5a of YA-DOS, a DECB 16K ROM replacement. This firmware supports a wide variety of physical media. More info and download at the following link:

<https://tinyurl.com/ycku6wk8>

Ron Klein created and shared a disk image that contains over 3,000 Color Computer virtual floppy disks which works with YA-DOS and Michael Furman's pyDriveWire.

Intro video: https://youtu.be/riNy1nA0_GI

Download image: <https://github.com/mrgw454/CCHDD>

John Guin noted that Microsoft had released the GW-BASIC product under the MIT license. He also provided the GitHub link:

<https://github.com/microsoft/GW-BASIC>

Nathan Byrd announced that he developed an online version of Dungeons of Daggorath. <http://daggorath.online/>

Guillaume Major announced the release of Coco SDC Explorer (SDCX) which can now execute BASIC programs in ASCII, as long as they have the BAS extension. It is available at the CoCo archive (ZIP file download): App: <https://tinyurl.com/ydetnyrl>

Source: <https://tinyurl.com/y72vspkf>

Ed Snider, aka Zippster, announced that he is getting ready for another run of his CoCo keyboard CoCoMECH. This time he is trying out black key caps. <https://tinyurl.com/y6uapup2>



Dave Philipsen announced that he is ready to ship the CoCoDEV computer. More info at the following link: <http://www.davebiz.com/wiki/CoCoDEV>



Aaron Doughty published a video on YouTube where he discusses Paul's Keys2USB adapter. Great source of information! <https://youtu.be/hulOoHlcmM>

Ron Klein mentioned that the new MAME build, v 0.222 has been packaged and is ready for updating the Raspberry Pi. He provided two links, one for the R-Pi3 and a second one for the R-Pi 4:

R-Pi 3: <https://tinyurl.com/y8anp464>

R-Pi 4: <https://tinyurl.com/ydh3xnpv>

Nick Marentes now has available Rally-SC for the CoCo. He posted the following image to promote this new game.



<http://www.nickmarentes.com/>

Richard Lorbieski announced that Darren Atkinson's CoCo SDC will now also be available through Boyson Tech starting on June 22, 2020. This means that we can get our CoCo SDCs either from Ed Snider or Boyson Tech.

<https://tinyurl.com/y9jo5abz>

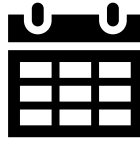


Tim Lindner announced that new revised user documentation for the CoCo SDC is now available at the following link:

<https://tinyurl.com/ybzzvgku>

The link points directly to the PDF file of the documentation.

Calendar of Events



Compiled by Bob Swoger

Glenside Color Computer Club, Inc. Business Meetings

Future dates: August 20th, September 17th, October 15th

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: Postponed until further notice
Location: To be determined

<http://glensideccc.com>



Image courtesy of D. Bruce Moore



A list of Coco web resources compiled by Steve Strowbridge

Source

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

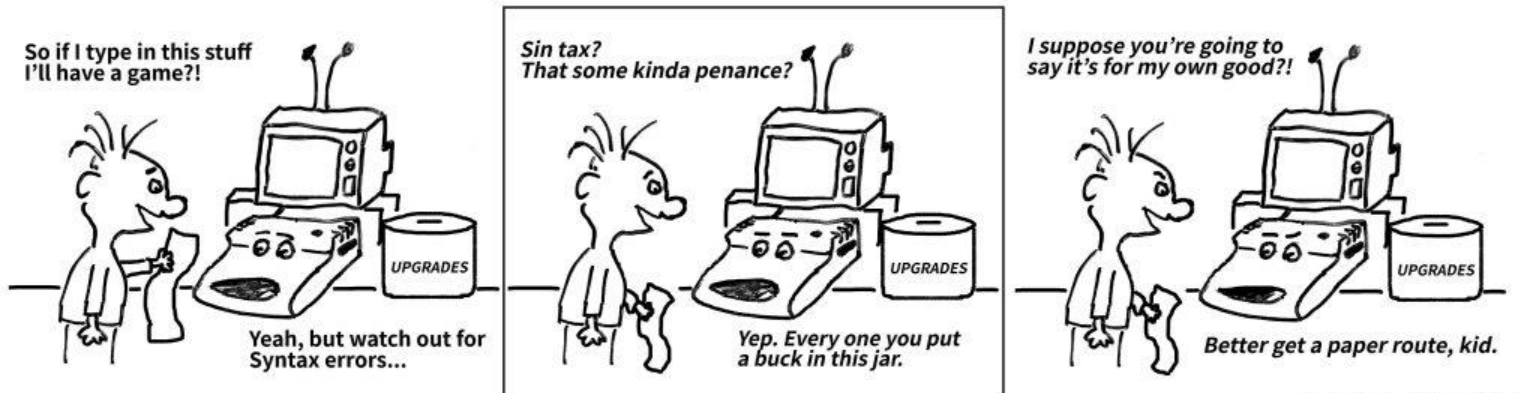
Link(s)

[Website](#) | [Classic splash page](#) | [Facebook](#)
[Facebook](#)
[Website](#)
[Facebook](#)
[Website](#) | [Facebook](#)
[Website](#) | [Facebook](#)
[Website](#) | [Facebook](#)
[Get Discord](#) | [Server Link](#)
[Website](#) | [YouTube](#)
[Website](#)

Close Parenthesis

D. Bruce Moore has kindly allowed CoCo~123 to publish his new CoCo themed comic. We now bring you comic strip #4 "Syntax Errors"

JOEY & COCO



(C) 2019 D. Bruce Moore