OCTOBER 1991

$2.00
CANADIAN $2.75

VOLUME 5
NUMBER 3

COLOR Computing

The bi-monthly magazine for Tandy Color Computer users

COCO PROGRAMMING LANGUAGES...

C

PASCAL

ASSEMBLY

BASIC

LOGO

COLOR Computing

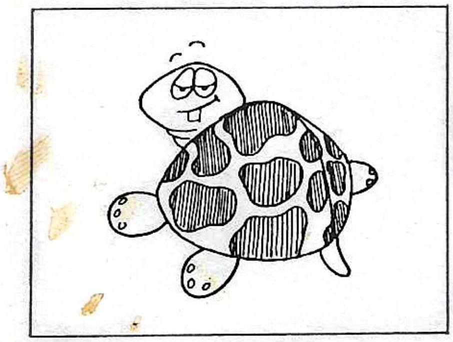
October 1991
Volume 5
Number 3

formerly TRS-80 Computing

TABLE OF CONTENTS

FEATURE ARTICLES

THE PROGRAMMER



PAGE 13

REVIEWS

COLUMNS/DEPARTMENTS

Color Computing is published bi-monthly (February, April, June, August, October, December) during the year. Subscriptions in the U.S. are $13 for one year and $20 for two years (In Canada: $15 for one year and $21 for two years). Please contact us for foreign subscription rates. Each issue is mailed by the 20th of the month published, unless there is a production delay. Please let us know ahead of time if there is going to be a change in your address.

COLOR COMPUTING, 65 OAK RD., CANTON, MA. 02021-2605 TEL.# (617) 828-7749

FROM THE DESK OF... THE EDITOR

(This editorial was actually written at the beginning of January, 1992.)

Finally! The October 1991 issue of Color Computing is here at last! Yes, we are late again for the second time. What's the excuse now? No, not personal or business problems: this time we experienced many delays in receiving submissions and dealing with our printing company. We lost a few promised contributions, and therefore it was a struggle to get this issue out. Because of this, we were not able to combine the topic of the August issue (Business/Home Finance) with the October issue as we originally had hoped for. Although, we were able to add in a few extras to make this issue a little bigger than normal.

The editorial topic of this issue is Programming Languages, and we did our best to relate to it. I wrote a short article on good 'ol Color Logo, and there are a couple of other odds and ends that you normally would not see in a regular edition.

There is one major issue that I would like to address to all of you before I close. I and many others, have noticed that the CoCo seems to be really dissipating from the community. Ever since Radio Shack discontinued everything for the CoCo a year ago, many seem to have lost a lot of interest and enthusiasm for their faithful computer. As a result, many third party vendors have folded, clubs have called it quits, and the few publications out have cut down tremendously. The 68xxx-based machines have taken over, and this puts a huge dent in the likelihood of Color Computing continuing. We have not made a final decision on what to do yet, but it looks as though we are going to have to cut down on pages and become quarterly rather than bi-monthly. Hopefully we'll get some good luck in the next couple of months so we'll be able to keep up with the pace. But for now, my only advice is: PLEASE KEEP SUPPORTING THE TANDY COLOR COMPUTER!

Joe Ahern

COLOR COMPUTING
SINCE 1987

EDITOR-IN-CHIEF & PUBLISHER:

JOSEPH W. AHERN

PROGRAM EDITOR:

DAVID McNALLY

CONTRIBUTING EDITORS

LEO BREHM
BOB VAN DER POEL
THOMAS SINCLAIR
MICHAEL HOLTRY
RAY KORNELE
DEREK SNIDER
TIKA CARR

COVER DESIGN BY DEREK SNIDER

Hacker's Fare
Poem written by Tika Carr

Oh woa is me.
Staring at a blank screen.
Everything is plugged in
But my disks won't spin.
I look at my CoCo with a pout.
There's something I have to check out.
Power down and pull the plugs.
Open the case and probe for bugs.
Is my CPU fried? Or my GIME?
All this circuitry makes me dizzy.
Press down on the GIME and try again.
Still the same response from my friend.
I prepare for the operation ahead.
Knowing my CoCo is brain-dead.

I remove the CPU and unplug the holes.
Put in a socket and solder it to hold.
Snap in a new chip and finish up.
Anticipating what I'll see when I power up.
I plug it all in and try again.
What a joy to know I haven't lost my friend!

NEW COCO PRODUCTS

NEW FOR '92 FROM SUNDOG SYSTEMS

1. WAR MONGER- wargame designer ($29.95)
2. GRAFEXPRESS 2.0- graphics/music programming environment for 128K CoCo 3 ($34.95)
3. PHOTON- new arcade game for 128K CoCo 3. available in December ($34.95)

* Sundog Systems. P.O. Box 766. Manassas. VA 22111

CF83Forth- a 1983-standard Forth operating system/programming language for the CoCo 2/3: $17
* BDS Software. P.O. Box 485. Glenview. IL 60025

1991 Arcade Pak- Includes CycleJump. Minigolf. and Caveman: $24
* J.T. Rawlinson Software. 660 Briar Hill Ave. #204. Toronto. ON M6B 4B7. CANADA

NEW OS-9/68000 PRODUCTS

Microware Basic- designed for use with OS-9 on 68000-based computers: $500
* Microware Systems Corporation. 1900 NW 114th St. Des Moines. IA 50322

Super Sleuth- examine and modify binary program files on disk or in memory: $100 + $5 S/H
* Frank Hogg Lab. 204 Windmere Rd. Syracuse. NY 13205

Flexelint Binary- multifile C program check: $495
* Windsor Systems. 2407 Lime Kiln Lane. Louisville. KY 40222

The Programmer

THE PROGRAMMER OCTOBER '91

This month...

Menu - Select AUTOCOPY
by Jim Dees

Please don't be mislead by the function of the presented program, "AUTOCOPY". The objective is to demonstrate some neat and unusual BASIC programming action.

Such as, a DIR will display the names of the files from a disk, but won't allow anything to be done with them. Check out lines 105 through 130. The result is a four column, seventeen row DIR display, on a CoCo-3 eighty column monitor. It not only displays a full disk on one screen, it puts the names in string variables. Once the names are in a variable you can do "things" with them.

In this program the names are used in a "menu select" action for the normal disk drive to disk drive COPY action. Concentrating on the point of selection, there are many ways to do it. A program could be written so each displayed file name would be preceded with a number, letter,etc. Then the selection would be made by inputting the number/letter, and pressing <ENTER>.

Another method is to move the cursor, using the arrow keys, from file name to file name. Such a method results in fewer mis-selection errors. However, the best is to highlight each file name and pressing <ENTER> to make the selection of the highlighted name. Even fewer selectionmistakes occur this way.

The highlighting method is more complex to program, but not impossible. Line 470 causes the highlight using ATTR2,4. Your monitor may work better with different numbers, but the results will be thesame, a highlighted file name. (CoCo 2 users could POKE 282,0 and ,1 to change "case", which gets the reverse characters.) The selection in the presented program is done with the arrow keys. The "up" and "down " keys are made to repeat via the POKEs in lines 480and 530. But the neatest is getting the highlighted choice to "wrap" from the top to the bottom of the list, as well as from edge to edge and bottom to top. This action is obtained in lines 470 through 580.

Though the solution was easy, it was hard for me to figure out how to get the "wrap" to work when there is less then four full columns. That is, if there is only a column and a half of file names, how can only file names be highlighted without highlighting blank spaces on the screen where file names could be.

The simple solution was to test to see if the variable has a name before printing it. If it doesn't, just go on to test the next variable; see lines 500, 525, 550 and 575.

The most perplexing problem was how to get COPY, or other commands like KILL, RENAME, SAVE, etc. to work from within a program. Try it, itsnot straight-forward. After a bunch of keyboard banging and then some thought (may have been better if I had tried doing it the other wayaround), I found the code in lines 590 and 595 would get the job done.

Last, I can't read my monitor's 80 column display, a composite monochrome. It took three of us, one being Bob van der Poel, to come up with the code in line 700. If you have trouble reading your display, try the code. If the resulting color or read ability doesn't suit, try changing the PALETTE and/or CLS numbers. Of course if you don't have the problem don't use the line.

So there it is for this time, a way to get disk file names in string variables, a way to move the cursor with the arrow keys, a way to highlight screen displays, a way to make keys repeat in BASIC, how tocause "wrap" to work cleanly even with an unknown number of display elements, and finally, a way to clean up the 80 column display if you have trouble. (I still can't get HPRINT to display cleanly. Anyone have any hints?)

<u>THE LISTING:</u> AUTOCOPY

```
1 ' ****************************
2 ' *    >>> "AUTOCOPY" <<<    *
3 ' *                          *
4 ' *     FOR A MONOCHROME     *
5 ' *     COMPOSITE/COCO-3     *
6 ' *                          *
7 ' *         (C) 1991         *
8 ' *   BY COLOR COMPUTING     *
9 ' ****************************
10 PCLEAR1:CLEAR10000:DIMFL$(68)
.DT$(68).CN$(12):L$="":WIDTH80:P
ALETTE 0.0:PALETTE 8.63:CLS1:ATT
R0.0:GOSUB86
11 GOSUB33:GOSUB37
12 '***
13 '*** Display a DIR ***
14 '***
15 CLS:GOSUB33:CX=0:RX=3:X1=0:X2
=0:C=0
16 FOR X1=3 TO 11:DSKI$SD.17.X1.
Y$.Z$:X$=Y$
17 FOR X2=1 TO 128 STEP32:DF$=MI
D$(X$.X2.11):DT$=MID$(X$.X2+11.1
):AN=ASC(MID$(DF$.1.1)):IF AN=0
THEN20 ELSE IF AN=255 THEN20 ELS
E C=C+1:FL$(C)=LEFT$(DF$.8)+"."+
RIGHT$(DF$.3):DT$(C)=DT$
18 RX=RX+1:IF RX=21 THEN RX=4:CX
=CX+20
19 LOCATECX.RX:PRINTFL$(C)::IF C
=68 THEN22
20 NEXTX2
21 IF X$=Z$ THEN NEXTX1 ELSE X$=
Z$:GOTO17
22 LOCATE0.21:PRINT">>> Drive"DN
"has"FR"FREE granules"::RETURN
23 '***
24 '*** Clear Program Array & Ch
eck For Empty Disk ***
25 '***
26 LOCATE27.10:ATTR0.0.B:PRINT"C
learing Array Values"::FOR X=1 T
O 68:FL$(X)="":NEXTX:ATTR0.0:C=0
27 FR=FREE(DN):IF FR=68 THEN LOC
ATE 27.8:PRINT"     *** Empty Dis
k ***      ":SOUND20.6
28 SOUND245.1:LOCATE27.10:PRINT"
Press <ENTER> to continue":
29 I$=INKEY$:IF I$="" THEN 29 EL
SE IF I$< >CHR$(13) THEN29 ELSE R
ETURN
30 '***
31 '*** Screen Header ***
32 '***
33 LOCATE27.1:PRINT"<<<  A U T O
 C O P Y  >>>":RETURN
34 '***
35 '*** COPY ***
36 '***
37 CLS:GOSUB33:LOCATE9.4:PRINT"C
OPY:   - Copies the selected pro
gram from one drive to another."
38 LOCATE17.5:PRINT"- Note: Pres
s <C> to input directly on the C
ommand Line."
39 CX=5:CN$=MID$(CN$(C).1.5)+CHR
$(34)+"program name:":GOSUB81'**
* Print Command on Command Line
40 LOCATE13.10:PRINT"Drive numbe
r of the SOURCE disk (0-3) ? ":
GOSUB80
41 IF I$="C" OR I$="c" THEN LOCA
TE33.21:PRINT::END
42 SD=VAL(I$):PRINTSD:IF SD<0 OR
 SD>3 THEN SOUND10.2:GOTO37
43 CN$=CN$+STR$(SD)+CHR$(34)+" T
O "+CHR$(34)+"program name:":GOS
UB82
44 LOCATE13.12:PRINT"Drive numbe
r of the DESTINATION disk (0-3)
?"::GOSUB80
45 IF I$="C" OR I$="c" THEN41
46 DD=VAL(I$):PRINTDD:IF DD<0 OR
 DD>3 THEN SOUND10.2:GOTO37
47 CN$=CN$+STR$(DD)+CHR$(34):GOS
UB82
48 IF SD=DD THEN LOCATE5.20:SOUN
D10.4:PRINT"BOTH DRIVE NUMBER CA
N NOT BE THE SAME !  Press Any K
ey to continue."::GOSUB80:GOTO37
49 CLS:GOSUB33:LOCATE8.3:PRINT"U
se the arrow keys to select the
program name you want to copy.".
" When you have highlighted the
program you want press <ENTER>."
:
50 GOSUB25:GOSUB15:GOSUB81:CX=0:
RX=4:C=1
51 LOCATECX.RX:ATTR2.4:PRINTFL$(
C):
```

```
52 I$=INKEY$:IF I$="" THEN52 ELS
E ATTRØ.Ø:LOCATECX.RX:PRINTFL$(C
);
53 IF ASC(I$)=94 THEN POKE341.25
5:RX=RX-1 ELSE58'*** Up Arrow
54 IF RX<4 THEN RX=2Ø
55 IF C=1 OR C=18 OR C=35 OR C=5
2 THEN C=C+16:GOTO57
56 C=C-1
57 IF FL$(C)="" THEN53 ELSE51
58 IF ASC(I$)=9 THEN CX=CX+2Ø EL
SE63'*** Right Arrow
59 IF CX>74 THEN CX=Ø
6Ø IF C=>52 THEN C=C-51:GOTO62
61 C=C+17
62 IF FL$(C)="" THEN58 ELSE51
63 IF ASC(I$)=1Ø THEN POKE342.25
5:RX=RX+1 ELSE68'*** Down Arrow
64 IF RX>2Ø THEN RX=4
65 IF C=17 OR C=34 OR C=51 OR C=
68 THEN C=C-16:GOTO67
66 C=C+1
67 IF FL$(C)="" THEN63 ELSE51
68 IF ASC(I$)=8 THEN CX=CX-2Ø EL
SE73'*** Left Arrow
69 IF CX<Ø THEN CX=6Ø
7Ø IF C<18 THEN C=C+51:GOTO72
71 C=C-17
72 IF FL$(C)="" THEN68 ELSE51
73 IF ASC(I$)<>13 THEN SOUND1Ø.2
:GOTO73'*** Program name selecte
d with <ENTER> key
74 CN$="COPY "+CHR$(34)+FL$(C)+"
:"+STR$(SD)+" TO "+CHR$(34)+FL$(
C)+":"+STR$(DD)+CHR$(34):GOSUB82
75 SD$=STR$(SD):MID$(SD$.1.1)=":
":DD$=STR$(DD):MID$(DD$.1.1)=":"
76 COPY FL$(C)+SD$ TO FL$(C)+DD$
:GOTO11
77 '***
78 '*** Command Line Display ***
79 '***
8Ø I$=INKEY$:IF I$="" THEN8Ø ELS
E RETURN
81 LOCATE5.23:PRINT"- Command Li
ne Instruction:";
82 LOCATE33.23:PRINT"
                     ";:ATTRØ,Ø.U
:LOCATE33.23:PRINTCN$;:ATTRØ.Ø:R
ETURN
83 '***
84 '*** Config Screen for 8Ø Col
umn and Colors
85 '***
86 WIDTH8Ø:PALETTE Ø.Ø:PALETTE 8
.63:POKE&HFF98.4:POKE&HFF99.&H75
:POKE&HEØ45.4:POKE&HEØ46.&H75:PO
KE&HFEØ5.23:POKE&HFEØ6.47:POKE&H
FEØ7.16Ø:CLS1:POKE&HFEØ7.Ø:RETUR
N
```

END OF PROGRAM

COMMON COMPUTER LANGUAGES

BASIC - Beginner's All-Purpose Symbolic Instruction Code

PASCAL - used in education and computer design

FORTRAN (FORmula TRANslation) - used in scientific work

COBOL (COmmon Business-Orientated Language) - used in business

LISP (LISt Processor) - used in work on artificial intelligence

PL/I (Programming Language I) - used in business and science

New Releases From Color Computing Software

The Sound Library Volume One

This new sound library is that extra touch you have been looking for. Now you can have digitized sound in your programs without the freeze effect. Sounds play independent of your program using the CoCo's multi-tasking abilities. The package includes five digitized sounds and an easy to read, three page instruction manual that shows you how to load the sounds into your BASIC programs.

CoCo 3 Only.........$6.95+$.75 S&H

Markbook Version 1.0

This is a complete grading program for school or for home. The program keeps track of 35 students in up to nine different subject areas. Each student may have up to 20 grades. The program can average one students' grades or all of the students' grades. A printout can be made of the list of averages. The program is all menu driven and very easy to use. The program contains one disk and instructions.

CoCo 1&2..........$9.95+$1.00 S&H

Find It! Word Puzzle

This new game will have you puzzled for hours. The computer will choose a word from its dictionary and you have to make as many other words as you can from the letters in the word. Did you know that you can make up to 148 words out of the word Christmas? The dictionary included gives the computer all the possible combinations, so you can't cheat! Try it! It's not easy.

CoCo 3 Only........$9.95+$.75 S&H

Using ROM Routines PART I

by David Mcnally

The CoCo has many ROM routines that can be used from BASIC. The only problem with these routines is that many people do not know how to EXECute them.

The program listing below shows you how to use the JOYIN ROM routine. The listing contains a lot of remarks to explain what is going on within the program. After you learn how to call it up, try using it in your own programs. In the next couple of articles I will explain how to call up the INKEY and DIR ROM routines. I will also explain how to use the DSKCON routine which allows you to read from and write to a disk.

<u>THE LISTING:</u> JOYIN

```
0 REM *************************
1 REM DEMO PROGRAM TO CALL THE
2 REM JOYIN ROM ROUTINE FROM
3 REM A BASIC PROGRAM.
4 REM COPYRIGHT (C) 1991
5 REM BY COLOR COMPUTING
6 REM *************************
10 CLS
11 REM JOYIN ROM ROUTINE STARTS
       AT &HA00A.  PEEK HERE FOR
       MOST SIGINIFICANT BYTE
12 A=PEEK(&HA00A)
13 REM PEEK AT START ADDRESS+1
       FOR LEAST SIGNIFICANT
       BYTE
14 B=PEEK(&HA00A+1):' &HA00B
15 REM MULTIPLY THE VALUE OF 'A'
       BY 256 AND ADD 'B' TO GET
       ROM ROUTINE EXEC ADDRESS
16 C=A*256+B
17 REM NOW CALL JOYIN ROUTINE
18 EXEC C
19 REM ROUTINE SAMPLES ALL FOUR
       JOYSTICK POTS AND STORES
       VALUES IN 'POTVAL' THRU
       'POTVAL+3' WHICH IS
       LOCATED AT &H15A-&H15D
20 REM READ LEFT JOYSTK UP/DOWN
22 LU=PEEK(&H15D)
24 REM READ LEFT JOYSTK RGHT/LFT
26 LR=PEEK(&H15C)
28 REM READ RGHT JOYSTK UP/DOWN
30 RU=PEEK(&H15B)
32 REM READ RGHT JOYSTK RGHT/LFT
34 RR=PEEK(&H15A)
36 REM PRINT RESULTS
38 CLS:PRINT"LEFT JOYSTICK UP/DO
WN: ":LU
40 PRINT"LEFT JOYSTICK RIGHT/LEF
T: ":LR
42 PRINT"RIGHT JOYSTICK UP/DOWN:
 ":RU
44 PRINT"RIGHT JOYSTICK RIGHT/LE
FT: ":RR
46 REM REREAD JOYSTICK VALUES
48 GOTO 18
```

END OF PROGRAM

ROAMING GHOSTS
BY DAVID MCNALLY

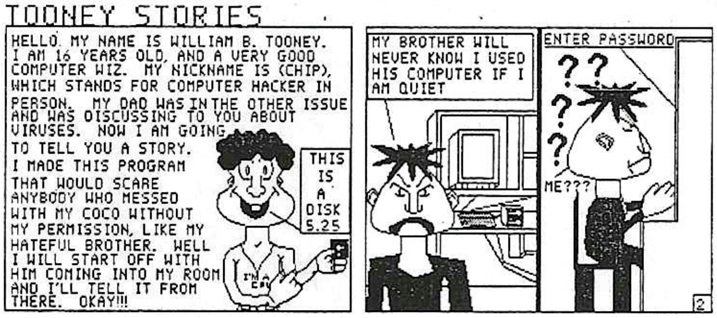
Mostly everyone thinks of Halloween when you mention the month of October. On Halloween night, not only do you go walking door-to-door dressed up in your scary costume looking for goodies, but you are also on the look-out for ghosts, witches, vampires, and all kinds of spooks of that nature. After a long night you come home to relax in front of your favorite computer to try to get away from it all. But to your surprise you spot another ghost in front of a nearby building.

THE LISTING: ROAMGHST

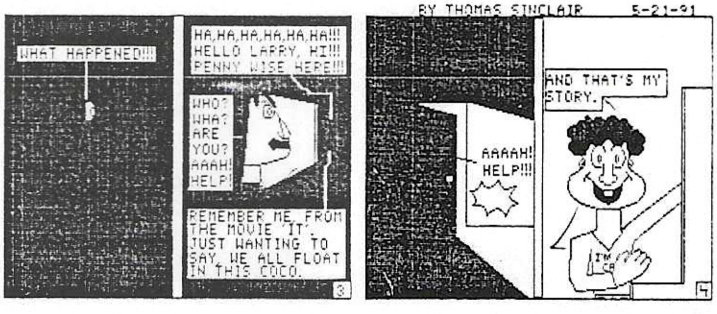
```
0 ' ROAMING GHOSTS
1 ' COPYRIGHT (C) 1991
    BY COLOR COMPUTING
2 PCLEAR 8:DIM BK(50.45)
4 PMODE 4.1
6 PCLS1:COLOR 0.1
8 LINE(0.176)-(256.176).PSET
10 LINE(0.148)-(256.148).PSET
12 LINE(60.148)-(24.176).PSET
14 LINE(116.148)-(80.176).PSET
16 LINE(180.148)-(144.176).PSET
18 LINE(236.148)-(200.176).PSET
20 LINE(32.0)-(32.148).PSET
22 LINE(0.16)-(12.16).PSET
24 LINE -(32.36).PSET
26 LINE(12.16)-(12.148).PSET
28 LINE(223.0)-(223.148).PSET
30 LINE(256.20)-(224.36).PSET
32 LINE(60.12)-(88.52).PSET.B
34 LINE(96.12)-(124.52).PSET.B
36 LINE(132.12)-(160.52).PSET.B
38 LINE(168.12)-(196.52).PSET.B
40 LINE(74.12)-(74.52).PSET
42 LINE(75.12)-(75.52).PSET
44 LINE(60.32)-(88.32).PSET
46 LINE(60.33)-(88.33).PSET
48 LINE(110.12)-(110.52).PSET
50 LINE(111.12)-(111.52).PSET
52 LINE(96.32)-(124.32).PSET
54 LINE(96.33)-(124.33).PSET
56 LINE(146.12)-(146.52).PSET
58 LINE(147.12)-(147.52).PSET
60 LINE(132.32)-(160.32).PSET
62 LINE(132.33)-(160.33).PSET
64 LINE(182.12)-(182.52).PSET
66 LINE(183.12)-(183.52).PSET
68 LINE(168.32)-(196.32).PSET
70 LINE(168.33)-(196.33).PSET
72 LINE(60.16)-(88.16).PSET
74 LINE(65.24)-(65,52).PSET
76 LINE(67.20)-(60.26).PSET
78 LINE(66.21)-(66.23).PSET
80 LINE(83.24)-(83.52).PSET
82 LINE(83.20)-(88.25).PSET
84 LINE(84.21)-(84.23).PSET
86 LINE(96.16)-(124.16).PSET
88 LINE(101.24)-(101.52).PSET
90 LINE(103.20)-(96.25).PSET
92 LINE(102.21)-(102.25).PSET
94 LINE(119.24)-(119.52).PSET
96 LINE(119.20)-(124.25).PSET
98 LINE(120,20)-(120.23).PSET
```

```
100 POKE 178.18:PAINT(4,24),,4
102 POKE 178.0:PAINT(20.28),.4
104 PAINT (236.40),.4
106 POKE 178.2:PAINT(0.0),.4:PAI
NT(232.0),.4
108 POKE 178.0
110 LINE (132.16)-(160.16).PSET
112 LINE(168.16)-(196.16),PSET
114 LINE(139.20)-(132.25).PSET
116 LINE(137.24)-(137.52).PSET
118 LINE(138.21)-(138.23).PSET
120 PAINT(0.188),.4
122 LINE(155.24)-(155.52).PSET
124 LINE(155,20)-(160.25).PSET
126 LINE(156.21)-(156.23).PSET
128 LINE(173.24)-(173.52).PSET
130 LINE(175.20)-(168.25).PSET
132 LINE(174.20)-(174.23).PSET
134 LINE(191.24)-(191.52).PSET
136 LINE(191,20)-(196.25).PSET
138 LINE(192.21)-(192.23),PSET
140 LINE(96.88)-(160.148).PSET.B
142 LINE(100.104)-(156.148).PSET
.B
144 LINE(128.104)-(128.148).PSET
146 LINE(129.104)-(129.148).PSET
148 LINE(124.128)-(124.130).PSET
150 LINE(132.128)-(132.130).PSET
152 CIRCLE(128.88).32.4..3..50,0
154 POKE 178.1:PAINT(40.0)..4:PO
KE 178.0
156 PAINT(128.84),.4
158 PMODE 4.5:GOSUB 194
160 A$="R3D1R6L6D1R6L4D1R6L6D1R6
L6D1R8L11D1R4BR3R1BR3R1D1L1BL3L1
BL3L4D1L2R16D1L16D1L2R20D1L9BL2L
9D1R9BR5R6BD1BL2L4BL5L9:"
162 B$="BD1BR2R8BR2R6D1R4L18D1R1
8D1R3L24D1R24D1L2L27D1R29BR4BD1L
5BL7L15BL5L5D1R5BR5R16BR6R5:"
164 C$="D1R5L5BL9L18BL8L5D1R5BR8
R19BR8R5BD1BL12L20D1R20D1L20D1R1
9D1L19D1R18D1L19D1R17D1L20D1R16D
1L21D1R15D1L22:"
166 X=RND(208):IF X<48 THEN 166
168 Y=RND(120):IF Y<22 THEN 168
170 PMODE 4.5
172 GET(X-17.Y-2)-(X+32.Y+43).BK
.G
174 D$="BM"+STR$(X)+"."+STR$(Y)
176 DRAW D$:DRAW A$:DRAW B$:DRAW
 C$
178 GOSUB 196
180 PMODE 4.1:SCREEN 1.1:PMODE 4
.5
181 PUT(X-17.Y-2)-(X+32.Y+43).BK
.PSET
182 Z=RND(4):DD=RND(15):IF Z=MM
THEN 182
184 ON Z GOTO 186.188.190.192
186 X=X+Z+DD:IF X>208 THEN X=X-Z
-DD:GOTO 182 ELSE GOTO 198
188 X=X-Z-DD:IF X<48 THEN X=X+Z+
DD:GOTO 182 ELSE GOTO 198
190 Y=Y+Z+DD:IF Y>120 THEN Y=Y-Z
-DD:GOTO 182 ELSE GOTO 198
192 Y=Y-Z-DD:IF Y<20 THEN Y=Y+Z+
DD:GOTO 182 ELSE GOTO 198
194 PCOPY 1 TO 5:PCOPY 2 TO 6:PC
OPY 3 TO 7:PCOPY 4 TO 8:RETURN
196 PCOPY 5 TO 1:PCOPY 6 TO 2:PC
OPY 7 TO 3:PCOPY 8 TO 4:RETURN
198 MM=Z:GOTO 170
```

END OF PROGRAM


TOONEY STORIES
HELLO. MY NAME IS WILLIAM B. TOONEY. I AM 16 YEARS OLD, AND A VERY GOOD COMPUTER WIZ. MY NICKNAME IS (CHIP), WHICH STANDS FOR COMPUTER HACKER IN PERSON. MY DAD WAS IN THE OTHER ISSUE AND WAS DISCUSSING TO YOU ABOUT VIRUSES. NOW I AM GOING TO TELL YOU A STORY. I MADE THIS PROGRAM THAT WOULD SCARE ANYBODY WHO MESSED WITH MY COCO WITHOUT MY PERMISSION, LIKE MY HATEFUL BROTHER. WELL I WILL START OFF WITH HIM COMING INTO MY ROOM AND I'LL TELL IT FROM THERE. OKAY!!!
THIS IS A DISK 5.25
I'M A
MY BROTHER WILL NEVER KNOW I USED HIS COMPUTER IF I AM QUIET
ENTER PASSWORD
? ? ? ?
ME???
2

BY THOMAS SINCLAIR 5-21-91


WHAT HAPPENED!!!
HA,HA,HA,HA,HA,HA!!! HELLO LARRY, HI!!! PENNY WISE HERE!!!
WHO? WHA? ARE YOU? AAAH! HELP!
REMEMBER ME, FROM THE MOVIE 'IT'. JUST WANTING TO SAY, WE ALL FLOAT IN THIS COCO.
3
AAAAH! HELP!!!
AND THAT'S MY STORY.
I'M
4

LOOKING BACK ON COLOR LOGO

by Joe Ahern

For those who haven't heard of`it before, LOGO is a graphics computer language which allows users (primarily children) to create colorful and interesting graphics on the computer screen. It can be used to do just about anything any other computer language can do.

Logo is a very simple language to learn, and that's what sets it apart from other computer languages such as BASIC or Pascal. Even a very young child can begin to do interesting and fun things with Logo after a five to ten minute introduction to the language. Most children never master BASIC or other computer languages because it requires hours of intensive study before anything enjoyable can be done on it. Logo is designed so that young children can learn how to program a computer without having to study at it. It will also retain a child's interest since it can be learned at many different levels.

Radio Shack used to sell two versions of Logo: Color Logo, the regular version of Logo, and Super Logo, an upgraded version which included decimal arithmetic and list processing.

Turtle Geometry

"Turtle Geometry" refers to Logo's graphics abilities. The Color LOGO turtle is a small triangular shape which appears in the middle of the computer screen. The child can draw pictures by giving simple commands which move the turtle from one spot on the screen to another. As the turtle moves, it draws a line on the screen.

The child must learn only four simple commands in order to start making designs. These commands are FORWARD (FD), BACK (BK), RIGHT (RT), and LEFT (LT). As the child gains experience with Logo, other capabilities of the language are discovered. The child learns how to change the backround color of the screen, how to change the color of the line drawn by the turtle, how to erase lines, etc. Shortcuts, such as the use of the REPEAT command, are also learned. REPEAT tells the turtle to repeatedly follow a set of commands for the number of times specified.

LOGO's Different Modes

Color LOGO has three modes: RUN, EDIT, and DOODLE mode. Most new users of LOGO begin in the RUN mode, where commands can be entered directly. The turtle immediately carries out the command. The EDIT mode is learned next, where a series of commands called a procedure are entered. After the entire procedure is written, the user goes back to RUN mode and runs the program there. The DOODLE mode is used for those who want to draw shapes and designs by simply using special keys on the keyboard.

Even though Radio Shack has discontinued Color LOGO for quite some time now, I'm sure it is still being used by CoCoers and some elementary school teachers. Educators still like to use LOGO because it better prepares students for advanced languages than BASIC does. I, myself, teach Color LOGO to a third grade class at an elementary school here in Canton. The program is run on a network of eight CoCo 2's. I teach the course once a week for about an hour, and the children thoroughly enjoy it.

SIMULATING COMMANDS

by David McNally

There are many different versions of BASIC and unfortunately not all versions have the same commands. In this article I would like to show you how to simulate some of the more common commands found on other computers. Some of these commands have already been added to the CoCo 3's vocabulary but will be useful for the CoCo 2.

1.) Routine: Routine to simulate the CSRLIN function
Function: Used to determine what line the cursor is on
Use: After typing RUN, use PRINT FNA(1) to call routine from a program.

Listing:

```
5 DEF FNA(X)=INT(((PEEK(136)*256+PEEK(137))-1024)/32)-1
```

2.) Routine: Routine to simulate POS function
Function: Returns @ position of cursor
Use: After typing RUN, use PRINT FNB(1) to call routine from a program.

Listing:

```
10 DEF FNB(X)=(PEEK(136)*256+PEEK(137))-1024
```

3.) Routine: Routine to simulate HEX$ function
Function: Returns hexadecimal value of the number inputed
Use: After typing RUN, enter the decimal number to be converted.

Listing:

```
20 DG$="0123456789ABCDEF"
22 PRINT "ENTER NUMBER";:INPUT N
24 IF N<0 THEN N=(65536+N)
26 H$="":I=N
28 Q=INT(I/16):R=I-Q*16:H$=MID$(DG$,R+1,1)+H$:I=Q
30 IF I>0 GOTO 28
32 PRINT H$
```

4.) Routine: Routine to simulate MOD function
Function: Returns remainder of N/M
Use: Assign N and M decimal values in a program before calling routine. Use PRINT FNMD(1) to call routine within a program.

Listing:

```
36 DEF FNMD(N)=(INT(N)-INT(INT(N)/INT(M))*INT(M))
```

COLOR Computing

SUBSCRIPTION FORM

COLOR COMPUTING, a bi-monthly magazine for Tandy Color Computer users, has up to 35 pages with programs, informative articles, product reviews, graphics, hints & tips, etc. Different issues feature games, utilities, business, beginner's guides, and always our end-of-the-year holiday issue. Just to show you that every other month brings you a variety for your Tandy Color Computer. Continue to support the Color Computer by subscribing to Color Computing today! (trial issues can be bought for $2.00 each)

U.S.A.	**CANADA**
I would like:	I would like:
___ 1 year (6 issues) for $13.00	___ 1 year for $15.00
___ 2 years (12 issues) for $20.00	___ 2 years for $22.00
___ Trial issue for $2.00	___ Trial issue for $2.75

I am paying by:

___ CASH ___ CHECK ___ MONEY ORDER (U.S. CURRENCY ONLY)
(Payment must accompany order. Sorry, VISA and MC not accepted.)

NAME ______________________________________

ADDRESS ______________________________________

CITY ________________ STATE ______ ZIP CODE __________

* COLOR COMPUTING is sent First Class Mail, and in yellow manilla envelopes for subscribers residing outside of the United States.
* We are not responsible for issues you donot receive after we send it, but we do check after the first issue to see if you got the magazine alright.
* Make all checks payable to COLOR COMPUTING.
* Subscriptions are non-refundable (we suggest you buy a trial issue before subscribing).
* If you live overseas, please contact us for subscription rates outside of the continent.

THANK-YOU!

COLOR COMPUTING
65 Oak Road
Canton, MA. 02021-2605

CUSTOMIZING MULTI-VUE

by Ernest Bazzinotti jr.

What we're going to do here is put the Multi Vue package on one disk and add some new features. Start by formatting three blank disks for forty tracks using your system master. Also have a backup copy of the Multi Vue disks available.

Using the boot config disk, make a new bootable disk and label it accordingly. Include the following device descriptors: P, T1, pipes, modem descriptors, all the 40 track options you want, 60 hz., term_win, all the windows, and a full set of commands.

With the system master in /d0 load the following commands: dsave, copy, and makdir. What your going to do here is copy the modules directory from Multi Vue to your bootable disk. Put the Multi Vue master in /d1 and the bootable disk in /d0. Type the following statements:

```
makdir /d0/modules <enter>
chd /d1/modules <enter>
dsave /d1 ! (-x chd /d0/modules) <enter>
```

(Note: See page 160 of the OS-9 manual for more details on this command.) Now type chd /d0/modules <enter>. Then type edit Bootlist.MV <enter>. Move down the listing and change all the 35 track descriptors to 40 tracks and then quit the editor.

Next reboot with the bootable disk and type chx /d0/cmds <enter>. Load the following commands: dsave, os9gen, makdir, copy. Label a blank formatted disk Custom Multi Vue and put this into /d1. Now type the following:

```
chd /d0/modules enter
os9gen /d1 < Bootlist.MV enter
```

When all this is finished put the boot/config disk into /d0 and type:

```
chd /d0/modules <enter>
```

Copy all three of the 40 track device descriptors to the module directory in /d1 and verify with dir /d1/modules <enter>. Using a backup copy place the Multi Vue master into /d0 and type the following statements:

```
chd /d0 <enter>
dsave /d0 /d1 ! shell <enter>
```

Note that you will have to reload the copy command after every dsaving operation. Repeat the above for side "B" of Multi Vue (And reload the copy command.)

Put the Custom Multi Vue disk in /d0 and edit /d0/sys/env.file and Build.MV. Change all the 35 track descriptors to 40 tracks.

Next type chd /d0/modules <enter>. Edit Bootlist.MV and change all 35 track device descriptors to 40 tracks. This was desaved once already and must be changed again. Verify with a dir /d0/modules.

At this point we can RUN BuildMV. If you get no errors you now have a customized/bootable disk. NOTE that before you run BuildMV make sure that the os9gen and copy commands are in memory: this will avoid any further mistakes. When running BuildMV there can be no errors or you will either wind up with a disk that won't boot or a fragmented disk.

Don't worry, we're almost home. Reboot the computer and bring up the hi-res screen (use the hi-res joystick interface if you have one).

With the system master in /d0 and the Multi Vue disk in /d1 copy the following commands to the commands directory of the Multi Vue disk: modpatch, edit, copy, dsave, makdir, xmode, cobbler.

Remove the system master and with the M.V. disk in /d0 type the following statements:

```
chd /d0 <enter>
chx /d0/cmds <enter>
xmode /p Baud =04 <enter>
(this sets the printer baud rate)
            04=2400 baud
```

Type: edit Patch <enter>. Here is the listing for this file.

```
E: modpatch -s
   L /d0
   c 14 00 03
   v
   L /d1
   c 14 00 03
   v
   ( then quit the editor )
```

Now type Patch <enter>. Put a 40 track blank disk in /d1 and type the following statements:

```
chd /d0 <enter>
cobbler /d1
dsave -s24 /d0 /d1 ! shell <enter>
```

Remove the custom Multi Vue disk and re-insert an OS-9 System Master that supports the pipe module. Load the following three commands: dsave, copy, and makedir. Remove the System Master and re-insert the custom Multi Vue disk. Type:

```
chd /d0 <enter>
dsave -s24 /d0 d1 !shell <enter>
```

Wait until the dsaving operation is finished and then remove all your disks and turn off the computer. Now remove the hi-res interface and plug in your joystick or mouse. Turn the computer back on and load your new Multi-Vue disk.

What you've done here is changed the printer baud rate, changed the stepping rate for your disk drives to 6 m/s, and finally you will not need the hi-res joystick interface to punch up the hi-res screens. Just goto the Tandy menu and click on the hi-res icon. One last thing before we leave, if you want, you can rename the "Autoex" file to "Multi-Vue". That will allow you to make some changes in the startup file.

That's it! Good luck!

The TRADING POST

P.O. BOX 3453, CARBONDALE, IL 62902

From E.Z. Friendly Software

Leonardo's Paintbox	$26.95
Super Comics+	$21.45
Picture Bingo	$19.95
Data Form III	$19.95
Letter Form III	$19.95
Math Games	$19.95
Keyboard Commander	$19.95
Jack Rabbit Story Writer	$24.95

From Forrest Enterprises

Inventory Manager	$24.95

From 5-Star Software

Baseball Stats	$24.95
Quick Stats Ball Scoreboard	$22.95
Ball Cards Database	$19.95
Check Writer	$19.95
Bankman Checkbook Keeper	$24.95
Envelope Writer	$17.95
Coco Labels	$19.95
Printing Calculator	$14.95
General Statistics	$24.95
Brainbusters Test Writer	$24.95
HI & LO-RES JOYSTICK ADAPTER	$24.95

Specify Coco 1, 2 or 3. Disk Only.
Send Check or Money Order. Add $3.00
S/H. Canadian & Overseas MO Welcome.

USING ADVANCED TOPICS IN BASIC

Column written by Ray Kornele

In my last article I showed you how to do a mechanical approximation of a function in CoCo ECB. This time I will show you how to make the program modify itself on a disk drive.

I will also show the same program in standard CoCo BASIC (SCB). The modify portion, which uses the disk, will still work since the merge command is in the disk controller ROM. The purpose is not to write a sophisticated program, but to show how it can modify itself. Therefore, beware, if you want to use such things as square root or trancendentals (SIN, COS, TAN, LOG, EXP, etc.) the program would have to be modified. Also, if you enter a function whose graph does not cross the zero line, the program will go into an endless loop.

```
Program                             Commentary

10 CLS:GOTO 110                     CLear Screen and GOTO modify
20 X=100:R=0                        Initialize parameters
30 DEF FN Y(X)=X^2-4                DeFiNe original FuNction
40 DEF FN DY(X)=(FNY(X+.000001)-    DeFiNe derivative FuNction
FNY(X.000001))/.000002
50 IF ABS(FN Y(X))>4E-7 THEN X=X    Solve for X when Y=0-(FN Y(X)/FN
DY(X)):GOTO50
60 PRINT"ONE SOLUTION IS";X         PRINT solution
70 IF R=0 THEN X=-100:R=1:GOTO50    Reset parameters and re-solve
80 PRINT"PRESS ANY KEY TO CONTIN    Ask for restart of program
UE"
90 EXEC44539                        Wait for keypress
100 RUN"FUNCTXB                     Restart the program
110 PRINT"YOUR ORIGINAL FUNCTION    Identify original function
IS         Y=X^2-4":REM ^=UP-ARO
120 INPUT"ENTER YOUR NEW FUNCTIO    Prompt for new function
N          (E.G. X^2-4)";Y$
130 CLS:PRINT"Y="Y$                 PRINT your new function
140 OPEN"O",1,"TEMP/BAS"            OPEN file for modification
150 PRINT #1,"10"+CHR$(13);         PRINT for delete line 10
160 PRINT #1,"30 DEF FN Y(X)="+Y    PRINT for change line 30$+CHR$(13);
170 CLOSE:MERGE"TEMP",R             MERGE modify program and run it
```

The merge command will only work with ASCII files. However, BASIC data files are always ASCII, so this works fine. (See note at end of article.) Here is the same program in SCB.

```
Program                             Commentary

5 CLS:GOTO 100                      The differences are as follows;
10 CLS:X=100:R=0                    Line 140 now sets up changes to line
20 GOSUB 170                          200 instead of 20. This is where
40 IF ABS(Y)>1E-6 THEN X=X-Y/DY:      the calculations for Y (which
GOTO 20                               replaces FN Y(X)) will be done.
50 PRINT"ONE SOLUTION IS";X         Lines 160-190 calculate DY (which
60 IF R=0 THEN X=-100:R=1:GOTO40      replaces FN DY(X)).
70 PRINT"PRESS ANY KEY TO CONTIN    Line 200 calculates Y (which
UE                                    replaces FN Y(X)).
80 EXEC44539                        Notice that lines 160 and 180
90 RUN"FUNCTCB                        gosub 200 for calculations. This
100 PRINT"YOUR ORIGINAL FUNCTION      corresponds to the FN Y calls in
```

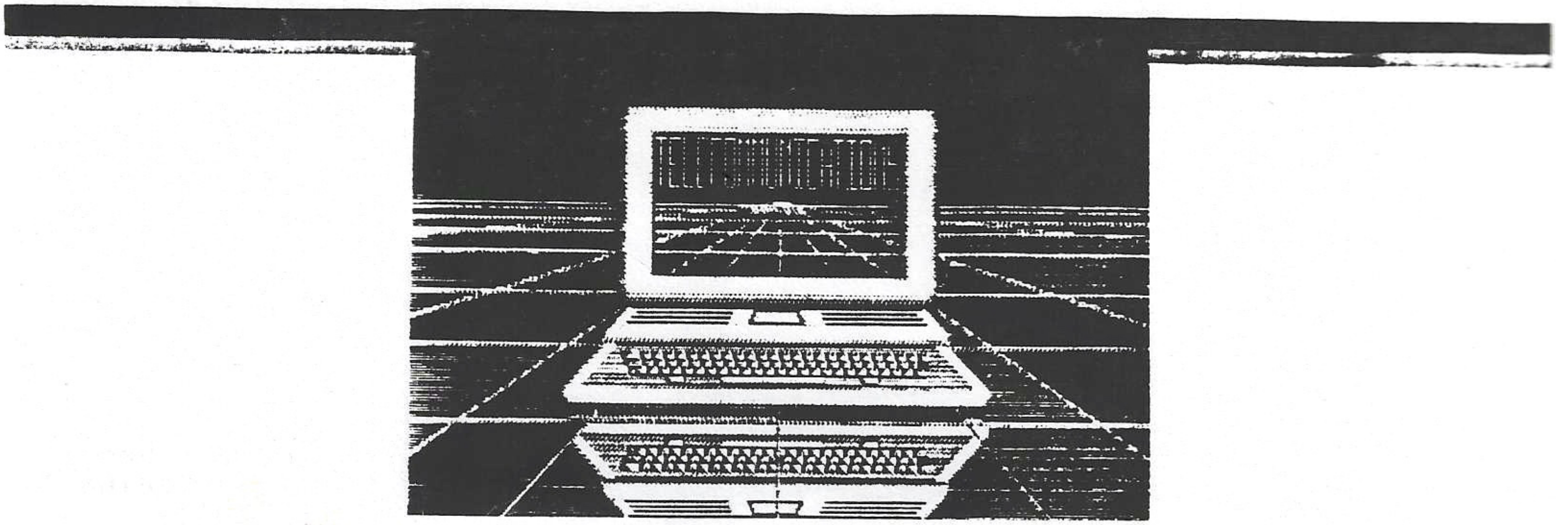
```
IS          Y=X^2-4":REM ^=UP-ARO
110 INPUT"ENTER YOUR NEW FUNCTIO
N          (E.G. X^2-4)";Y$
120 OPEN"O",1,"TEMP/BAS"
130 CLS:PRINT"Y="+Y$
140 PRINT #1,"5"+CHR$(13);
150 PRINT #1,"210 Y="+Y$+CHR$(13
160 CLOSE:MERGE"TEMP",R
170 X=X-.000001:GOSUB210
180 Y1=Y:X=X+.000002
190 GOSUB 210:X=X-.000001
200 DY=(Y-Y1)/.000002
210 Y=X^2-4
220 RETURN
```

the FN DY calculation. However, instead of FN Y(X-.000001) and FN Y(X+.000001), we must actually manipulate X itself. This is a global variable instead of the local variables which are used in the FN calls in ECB.);

Note: When entering your new functions don't use implied multiply. Instead of the algebraic expression 5X you must write 5*X. CoCo doesn't recognize implied multiply.

THE WORLD OF TELECOMMUNICATIONS

Column written by Michael Holtry



Hello there! Hope this article finds you all as CoCo crazy as I have been for years.

Most of the available online information services are both user friendly and computer friendly. Any computer should have no problem communicating with any other computer. As long as you have the right software and a modem that is compatible. You should be off and running without any problems.

The main job of a modem is translating data. As a modem sends data, it translates the Binary signals (the way that your computer communicates within itself) to Analog signals (or sound). It sends these signals over the phone lines to another computer. The modem on the receiving end converts the signals back to Binary signals.

Telecomputing involves several elements: data, data terminal equipment, data communications equipment, and a communications link (the telephone line). Other things are involved of course, but these are the most important ones!

Here are some vocabulary terms that you should be familar with:

Computer Data- any information that the computer can read in and use. Such information can consist of executable programs, text files, messages, graphics images, and so on.

Data Terminal Equipment- the computers involved in telecomputing. A terminal is only a place where the telecommunications journey begins and ends.

Communications Connections- computer to computer communications can take place in many different ways. Hooking two computers together with a null modem cable, to connect their serial ports directly,, is one way. Another and the most usual means of communications is via telephone lines. But ordinary phone lines cannot handle the computers binary format. A modem must be used to translate the binary data to a analog signal. It (the data) can now be easily transmitted over the phone lines. This is also a digital telephone network for computers and modems that run at a high speed, and require extreme accuracy! The telephone link doesn't consist solely of the telephone lines. There are microwave transmitters, satellite up and down links, electro- mechanical, electronic, and computer switching involved.

Data Communications Hardware- is just the modems and the interfaces, cables, or connectors.

A few of the modems jobs are as follows: 1--- Establishing and maintaining a communications link. 2--- Converting data from binary to analog format and the other way around also. 3--- Sending and receiving the data over the communication link.
The serial interface is used between the modem and computer. This interface is known as the RS-232C (An industry standard. So, let's take the mystery out this this "RS-232C" standard. The RS-232C is the standard as to how a serial port should communicate physically. It was established as a standard in 1969. This is how it brakes down and what each part of the designation stands for:

RS--- Acronym for Recommended Standard
232--- Id number for this certain standard
C--- for the latest revision of that standard

The RS-232C standard covers the electrical, mechanical, physical make ups of the interface, and functions of each signal. One of it's jobs is being a data conduit. But that is a small part of the functions that it performs. Some of these are organizing parellel data bits transmitted from the computer into a serail form, checking the parity, and removing and adding start and stop bits, flow control, and many other things besides

You should check your modem and computer manuals to see what type of modem cable you will need to make or purchase. With our CoCo's, it is either a DIN plug to a DB-25 cable. The Din plug end plugs into the CoCo serial port, and the DB-25 into the serial port of the modem. Or a DB-25 to DB-25 cable, one cable plugs into serial port of modem, and the other end connects into the RS-232C pack in the multi-pak. Both types are available, at a modest cost, from either a Color Computing or Rainbow magazine advertiser. By this time you should be able to visualize groups of 0's and 1's zipping around inside your computer, and doing this when instructed to do so.

So what does the modem do with the data that the computer sends to it? Well, the data normally leaves the computer through the serial port. The data is conducted to the modems serial port. Meanwhile the modem has been keeping a communications link open, checking for incoming data on that link, and watching for commands or data from the computer hooked to it. When the data arrives at the modems serial port, it has several jobs to do. 1--- Translating data from digital to analog format. 2--- Sending the data over the communications link. When data arrives over the communications link (Telephone lines) it 3--- is translated from analog to digital format, and 4--- it then sends data to the computer. The modem also handles error checking and flow control. More often as not though, the software handles the error checking and the such. Data translation and transmission is the modem's primary job though.

Why use an Analog signal instead of a digital one? Because the telephone systems and all the support hardware was designed for voice communications. This means that limited frequencies are available. These bandwidth's do not use all frequencies required by digital data. Some of the filtering and amplifying circuits sometimes cut off part of the upper end of a digital square wave signal. And the poor quality of the telephone lines make for variations in signal waves necessary. The modem then changes the square signal waves (digital) to one that the telephone networks can handle.

After the modem converts the signal from digital to analog, it is very easy to transmit it over the telephone lines. A signal is put on the phone line carrier wave using a process called "modulation". This works much the same as radio transmissions do. A carrier wave is a tone transmitted on a phone line. It is a tone of constant frequency and strength (a sine wave). A sine wave is a periodic electro-mechanical impulse that carries infomation on the telephone line. Modulation is when a signal is placed on a carrier wave, by changing one or more of it's characteristics. These characteristics include phase, frequency, or amplitude. Several different combinations are used in the higher speed modems. By the way, the word "MODEM" is a combination of the full name of the device. MODulating/ DEModulating, which is exactly what the modem does.

Like the serial port (RS-232C), the modem uses standards too. These standards are the Bell 103 (300bps), the Bell 212 (1200bps), and the CCITT V.22bis (2400bps). Virtually all information services, BBS's in America and Canada use these standards. In european nations different standards are in use.

There are Synchronous and Asynchronous modems on the market. But the Asynchronous one is what almost all information services, BBS's, etc. use in this, and a lot of other countries around the world. It is a data transfer mode that notifies a receiving computer system when each character begins and ends by framing bytes with additional bits. These extra bits are called start and stop bits. The start bit does just what it says. It tells the receiving computer system that a new byte is to follow. The stop bit tells the system that it is at the end of the last byte sent.

Both modems must agree on several important settings (parameters) before these systems can communicate. These parameters settings are the speed, what kind of duplex setting, start and stop bits, parity, and data bits. All these must match on both computer systems or it is a no go on communications. So you need to find out what the system that you are calling, has it's parameters set at. The first time that you call, if you get garbage on your screen then you have one of the parameter settings wrong. Just hang up, and rest everything as many times as you need to until you get a clear log on screen. Or call a friend who has been into telecommunications for a while, and have him set your terminal program parameter settings to the correct ones. This last one is the most simple, as you could be trying settings for quite awhile! While you are at it, get BBS and information services phone numbers from the same person that helped you set up your parameters setting. There are thousands of BBS's and several really good information service to choose from. When you are getting phone numbers for these, make sure to ask if it is a BBS that supports the CoCo, and has files for our loveable computer!

Well, that is all for this one, so take care. And keep on modeming!

COLOR Computing

BACK ISSUES ORDER FORM

Please send me:

JUNE 1987 PREMIER	$.60	FEBRUARY 1990 BEGINNERS	$1.50
NOVEMBER 1987	$.80	APRIL 1990 UTILITIES	$1.50
JUNE 1988	$.80	JUNE 1990 ANNIVERSARY/GAMES	$1.75
AUGUST 1988	$.80	AUGUST 1990 BUSINESS	$1.75
DECEMBER 1988 HOLIDAY	$1.00	OCTOBER 1990 HALLOWEEN	$1.75
FEBRUARY 1989 BEGINNERS	$1.00	DECEMBER 1990 HOLIDAY	$1.75
APRIL 1989 UTILITIES	$1.00	FEBRUARY 1991 BEGINNERS	$1.75
JUNE 1989 ANNIVERSARY/GAMES	$1.25	APRIL 1991 UTILITIES	$2.00
AUGUST 1989 GRAPHICS	$1.25	JUNE 1991 ANNIVERSARY	$2.00
OCTOBER 1989 HALLOWEEN	$1.50	OCTOBER 1991 PROGRAMMING	$2.00
DECEMBER 1989 HOLIDAY	$1.50		

TOTAL ENCLOSED $_____

(+ $1.00 FOR THE FIRST ISSUE AND $.35 EACH ADDITIONAL ISSUE FOR POSTAGE COSTS.)

NAME____________________________
ADDRESS_________________________
CITY_______STATE_____ZIP________

Allow up to 3 weeks for delivery.
Make all checks and money orders
payable to : COLOR COMPUTING

FOR THOSE WHO WERE WONDERING...

The Tandy Color Computer has one of the best designed microprocessors ever made. The Motorola 6809 chip is the direct ancestor of the 68000 series of chips found in the Macintosh and the Amiga computers. Like its more expensive cousins, the CoCo supports high resolution graphics, real time audio, and true multi-tasking operating systems.

The CoCo is not an 8-bit computer, rather it is one of the very first computers to use 16-bit addressing. For the novice the CoCo is ready to run with MicroSoft BASIC built in. For those focused on mastering the most complex problems, the CoCo supports PASCAL, C, Fortran, Assembler, Logo, and other languages.

The Tandy Color Computer has all the game playing capabilities of Nintendo as well as being a complete computer.

(This article originally appeared in a CoCo Club newsletter, and is written by Chris Johnson.)

COCO
PRODUCT REVIEWS

Software Review

Crystal City

The Crystal City, from Gosub Software, is a high speed action game for the Tandy Color Computer 3. It utilizes the CoCo's graphics and sound capabilities to produce non-stop action. As a pilot, you must fly through six levels of play destroying anything in your way (except the parachutes that contains precious fuel). This task is very difficult to do because of the high speed at which you are flying. Every level becomes more intense, and to make matters worse, you must win a one-on-one dogfight with the enemy to pass to the next level. At the end of each level you are rewarded with an amazing fireworks display. Be warned action game players, this one will take you a while especially having to survive the entire game with only one life.

The Crystal City comes on a two disk set. The first disk is a copy-protected disk containing the actual program which can be loaded by simply typing DIR. The second disk is an unprotected disk containing the data for the six different levels and the high scores. Backups of the second disk are highly recommended. A five page manual is included which drags out the story behind the Crystal City. The program is so easy to use that this manual is almost unnecessary.

The program as a whole is very well written (although I think it is a little overpriced). The sound is outstanding and the graphics aren't far behind. The only thing I didn't like about the program is that the ship's gun can not fire repeatedly by holding down the fire button. By the end of level two, you can't feel your your trigger finger. Thank God a pause option is included!

GRAPHICS: ****1/2
FRIENDLINESS: ****1/2
MANUAL: ***
VALUE: ***
OVERALL: ****

(SUNDOG SYSTEMS (DISTRIBUTED BY GOSUB SOFTWARE), P.O. BOX 766, MANASSAS, VA. 22111;TEL #(703) 330-8989;$34.95 + $2.50 S/H)

- David McNally

COCO LABELS

CoCo Labels is a mailing label program for those of you who just want a simple print-out of your mailing labels. The program does not have any special effects such as fancy fonts and graphic output. It is just an ordinary program for someone who wants to use it without any hassle. The program is easy to set up. There are two, easy to answer, questions required for the configuration. After answering the two questions you will need to RUN "BOOT" to load the program.

Once you have entered all of your labels, you may want to save it to be on the safe side. Once you have saved it, you may view your labels and edit any of them that you might have made an error in. If you decide you don't want a certain label, then simply delete it. Now, the printer should be turned on and the baud rate set to the configuration you set it for earlier in the program. When you are ready to print, you must choose from any 3 options given. Number one, is to print all the mailing labels you entered into the program. Number two, is to print only those mailing labels you want printed at that time. Example: You have 35 labels to print but you only want to print those between the numbers of 15-25, so all you do is let the computer know that, and it will only print those between 15-25. Number three, is to change simple font styles of the label. You have six of them from which you can choose from. They are normal print, underline print, elongation print, condensed print, elite print, and bold print. Once you are done printing you can clear the memory for a whole new bunch of labels, save the labels you just printed, load labels you had saved, or exit the program.

The program includes one single-sided diskette with four files on it: a configuration file, boot file, title file, and the label program itself. It also includes a six page manual that explains how to use the program in a step-by-step manner. The documentation also tells you the most common errors you may see in the program, if any. It requires a TANDY printer or one using similar codes to be able to print labels. I would say that this program is worth the money for anybody who wants a user-friendly mailing label program.

GRAPHICS: NONE
FRIENDLINESS: ****
MANUAL: *****
VALUE: *****
OVERALL: **** 2/3

(THE TRADING POST, P.O. BOX 3453, CARBONDALE, IL 62902:$19.95 + $3.00 S/H)

- Thomas Sinclair

OS-9 Software!

The Warped Drive Presents ...

Goodies Disk One

A collection of Basic09 programs in executable and source form. Both useful and fun, the programs will also help teach OS-9 programming.

> “... an irresistible temptation. ... you can't possibly go wrong with this purchase.” -Feb '91 Rainbow

Goodies Disk Two

Bigger and better, Disk Two demonstrates more advanced programming techniques than those used on Disk One. As before, both executable and source forms are included. (Disk two contains some ASM source code in addition to Basic09.)

Your choice: only $12ea. ($14CDN)
Complete catalog: $1 (refundable)**

**(Equivalent credit applicable to future purchases)

The Warped Drive

54 Martindale Drive NE
Calgary, AB
T3J 2V4

THE SHELL

A bi-monthly column devoted to the OS-9 operating system

This issue I am going to share a short, but very useful program. It was written in the middle of the night a few months ago when I decided that a number of the function names I'd chosen to use in a program were a bit too cryptic. I could have used a text editor or word processor to load in each file and make the changes; but there were over 25 files involved and about a dozen different changes to be made. What was needed was a program to read a file and replace each occurance of a specified string with a replacement. After that program was written and debugged (in which time I could have done the job with an editor!) the whole procedure was automated with a script file (the majority of which was created by doing block copies, etc. with the VED text editor). The program, replace, is presented here for your use. Note the use of a #ifndef to include the function findstr() if you are not using the 68000 version of OS9. This version of findstr() is not the most efficient, but it is a good example of how you can use existing library functions like index() and strcmp() to duplicate more complex functions you may find in other people's code. Replace is a filter--it really doesn't know anything about files. It just takes a stream of text from the standard input path, makes any needed changes, and pumps it out the standard output path. The command line:

```
   replace woof bark <myfile >newfile
```

would read the file "myfile", change all occurences of "woof" with "bark", and write this to the new file "newfile". Here is a little section of the script file which was used:

```
    load replace
    tmode .1 -pause
    replace <procs.c >temp "abc(" "makechange(";del procs.c;rename temp procs.c
    replace <form.c >temp "abc(" "makechange(";del form.c;rename temp form.c
    ....
    replace <procs.c >temp "del(" "deleteword(";del procs.c;rename temp procs.c
    ...
```

As you can see, there are a lot of temporary files created and the entire file was quite long. Not only that, but the process of making all the changes took quite a while--but with things being automated I just left the computer to do it's job while I had a nice dinner with some friends. When I got back, the job was done.

If you have any comments or suggestions on topics for "The Shell", drop me a note at:

P.O. Box 355
Porthill,ID
83853 USA

P.O. Box 57
Wynndel, BC
VOB 2NO, Canada

```
/* Replace.c

    This program will replace all occurances of <str1> with <str2> in
    the input file and send it to the output stream.

    (C) Copyright 1991
    Bob van der Poel Software
```

```
#include <stdio.h>

main(argc,argv)
int argc;
char *argv[];
{

   register char *oldstr,*newstr,*p;
   register int t,oldsize;

   char buff[520];

   if(argc!=3) terminate(0);

   if(argv[1][0]=='-' || argv[2][0]=='-') terminate(0);
   oldstr=argv[1];
   newstr=argv[2];

   oldsize=strlen(oldstr);

   for(;;){
      if((fgets(buff,sizeof(buff),stdin))==0) break;
      p=buff;

      for(p=buff;*p;)
      {
         if((t=findstr(1,p,oldstr))==0) break;
         t--;
         *(p+t)=0;             /* terminate part of str before target */
         fputs(p,stdout);      /* and send it */
         fputs(newstr,stdout); /* send replacement */
         p+=(t+oldsize);       /* point past target */
      }
      fputs(p,stdout);   /* send anything left after the target */
   }
}

terminate(err)
{
   if(!err){
      fputs("Usage: Replace <oldstring> <newstring>\n", stderr);
      fputs("       Changes all occurrences of <oldstring> with\n",stderr);
      fputs("       <newstring> in stdin. Sends new file to stdout\n",stder);
   }

   exit(err);
}

#ifndef OSK
```

```
/* This is a findstr() replacement for OS9-6809 users. */

#include <string.h>

findstr(startpos,str,target)
int startpos;
char *str, *target;
{
   register char *r;
   register int pos=0;

   if(r=index(str+startpos-1,*target)){
      if(strncmp(r,target,strlen(target))) pos=r-str+1;
   }
   return pos;
}

#endif

   <<< eof >>>
```

6809 ASSEMBLY LANGUAGE

By: Derek Snider

Here we are... Together once more to dig deep into the realm of the CoCo! If you were with us last time, you will remember our little talk about 6809 commands and registers. If not, I suggest that you get your hands on a copy, or else start flipping through a CoCo assembly language manual of some sort to try and catch up. Of course, if you have already spent a few days and nights hacking away with a 6809 assembler, then you may be able to keep up with us...

As I said before, and I now stress again... machine language has but two advantages...

SPEED and POWER!

Remember if you will, the sample programs we carefully typed into our assemblers, and followed the step by step instructions to assemble the code, save it, and execute it. And when we typed in that EXEC, how wide our eyes grew as the screen filled faster than we could blink.

Those of us that like to tinker away on our CoCo keyboards, slowly piecing clever bits of programming into nifty creations, those of us who would rather make a new game than play an old one, are now realizing how we could spice up our old programs with these simple little routines! But what if I told you that these routines were mere child's play? Surely you wish to do better... to make powerful routines to do much more impressive and wonderful things? Read on, my dear CoCo enthusiast...

For the next few examples, we will stick to the text and graphics screens. Why, you ask? Well, for one, you will see your results much more quickly, (so that you won't become frustrated and stop reading my column...) secondly, fancy tricks on the screen seem to prove more impressive to most people than number crunching, and lastly, they are usually simple to do.

Now for those of you who have used Greg Miller's, "GREG-E-TERM", will find this program's results to be very familiar. Nevertheless, you should find it interesting. This program will keep increasing the value of each character on the text screen, until every one has reached $FF, (255), an orange block. For this program, we will be using some new commands...

"INC", this command will increase either the "A" register, (INCA), the "B" register, (INCB), or a memory location, (ie: INC $1000).

Next there is "CLR", which works similar to "INC", because you can clear the "A" register, (CLRA), the "B" register, (CLRB), or a memory location, (ie: CLR $1000).

Another useful command is "COM", which will complement all the bits of a byte, (all the ones become zeroes and vice-versa). This command also can be used on the "A" register, (COMA), the "B" register, (COMB), or a memory location, (ie: COM $1000).

And last, (for now), but not least, "LEA". This command, (Load Effective Address), is used for incrementing or decrementing the 16-bit registers. It is much more powerful than "INC", and can be used to load a 16-bit register with the value of another 16-bit register, with an offset. To increment the "X" register by one, you would use "LEAX $01,X", which means "Load X with one greater than X". Therefore, "LEAX ,Y" would load the "X" register with the value in the "Y" register. Strangely enough, doing this is slightly faster than using "TFR Y,X".

"TFR" is the 6809 command for transferring the contents of one register to another. Remember... You can only transfer 8-bit registers to 8-bits registers, and 16-bit registers to 16-bit registers.

A command similar to "TFR", but much more powerful is "EXG". This command will exchange the contents of two registers with each other. A very useful command indeed!

Now on with the program! The routine will start at the beginning of the text screen, ($400), and get a character. If the character is less than $FF, it will increase the value of that location by one, and set the "B" register to $FF. This will continue until the program reaches the end of the text screen, ($5FF), and then check the "B" register. If the "B" register does not equal $FF, then the job is done. If it does equal $FF, then the "B" register is cleared, and the program starts back at the top of the screen again, and repeats the process. Now to see this fast little program go, type in FIG.1, save it, and assemble it.

If you have a CoCo 3, try the speed up poke, and see your CoCo really fly with speed. If you don't know the speed up poke, it's

POKE 65497,0.

To bring your CoCo 3 back to normal, type

POKE 65496,0.

For you CoCo 2 owners, you can use

POKE 65495,0

to speed up your machine, and

POKE 65494,0

to bring it back back to normal.

WARNING! - Do not do ANY disk access while your CoCo is under the influence of the speed up poke!

Many a time I've accidentally saved a program while in high speed mode, and wiped out my whole directory! Sometimes the data can be recovered if you know what you are doing, but it is not worth the hassle, so be careful! (There are programs available that allow you to do disk access while in double speed mode on a CoCo 3. FDOS by C. Fike is one, and there is also a 'patch' by Steve Bjork that I've seen around. Both are 'freeware' so you should be able to find them on local BBS's).

Derek Snider has been programming for over eight years on the CoCo. In this time he has become quite proficient in BASIC and 6809 assembly language. After developing a few BBS programs and simple games, he decided to start his own software company. And that company is D.S.D. Software.

With the CoCo 3, there is no need to reverse the actual data on the screen to reverse the colors, you just change the palette colors around. But on the good ol' CoCo 1 and 2, life wasn't so easy. Now even if you have a CoCo 3, this program is a good exercise, and the concept can be used as a simple encryption. As you know, if you reverse something, then reverse it again, it will be the same as it was in the first place.

So if you take some data and reverse it, it will seem like garbage. Reverse it again, and it will be normal. Simple but effective, (Sneaky eh?) Now type FIG.2 into your favorite assembler, (or FIG.3 if you are not using a disk drive controller).

When you're done, compare this to that old "PUT (X,Y)-(H,V),A,NOT" routine! If you are not familiar with the "PUT" routine, type in the BASIC listing of FIG.4 and see the difference. One thing I forgot to mention before... Don't forget that you need a "PMODE 4" and a "SCREEN 1,1" before you can see anything on the PMODE graphics screen... If you are not familiar with BASIC, then you probably aren't ready for machine language!

That's all we have time, (or room) for until October... So keep plugging away with your assemblers, and see if you can use what you now know to make more simple, but useful routines for yourself!
See you in December, where we will do some Christmas-related graphics, and continue to study the power of the CoCo!

- Derek Snider

FIG.1 Program: TEXTITUP.ASM

```
00100   * Program to increase the values of
00110   * all the characters on the text
00120   * screen until they all equal $FF
00130
00140           ORG     $3F00     Start program at $3F00
00150   START   LDX     #$400     Load X with start of screen
00160           CLRB              Clear B register
00170   LOOP    LDA     ,X        Load A with a character
00180           CMPA    #$FF      Does A=$FF?
00190           BEQ     SKIP      If yes then branch to SKIP
00200           INC     ,X        Increase value of character at X
00210           LDB     #$FF      Load B with $FF
00220   SKIP    LEAX    $01,X     Add $01 to X
00230           CMPX    #$600     Does X=$600?
00240           BNE     LOOP      If not branch to LOOP
00250           CMPB    #$FF      Does B=$FF?
00260           BEQ     START     If yes then branch to START
00270           RTS               Return control to BASIC
00280           END     START     End.  Set EXEC address.
```

FIG.2 Program: INVERT.ASM

```
00100   * Program to 'INVERT' the PMODE 4
00110   * graphics screen... FAST!
00120
00130           ORG     $3F00     Start program at $3F00
00140   START   LDX     #$E00     Load X with start of screen
00150   LOOP    COM     ,X+       Compliment value at X, and increment
00160           CMPX    #$2600    Does X=$2600?
00170           BNE     LOOP      If not then branch to LOOP
00180           RTS               Return control to BASIC
00190           END     START     End.  Set EXEC address.
```

FIG.3 Program: INVERT.ASM
(Non-disk version)

```
00100   * Program to 'INVERT' the PMODE 4
00110   * graphics screen...FAST!
00120
00130           ORG     $3F00     Start program at $3F00
00140   START   LDX     #$600     Load X with start of screen
00150   LOOP    COM     ,X+       Compliment value at X, and increment
00160           CMPX    #$1E00    Does X=$1E00?
00170           BNE     LOOP      If not then branch to LOOP
00180           RTS               Return control to BASIC
00190           END     START     End.  Set EXEC address.
```

FIG.4 Program: INVERT.BAS
(BASIC version)

```
10 PMODE 4:SCREEN 1,1              Set up graphics screen display
20 DIM A(6)                        Set up array for PUT command
30 FOR Y=0 TO 191                  Start a LOOP from top to thebottom
40 PUT(0,Y)-(255,Y),A,NOT          Invert a line using NOT.
50 NEXT Y                          Complete LOOP
60 END                             End.
```

Quality OS9 Level 2 Software

from

ColorSystems

WPShel – A Word Processing Oriented Graphics Shell
(Editor/Formatter/&Spelling Checker NOT included!)
$22 complete. (Add *$3* for overseas shipments.)

GAMES:

Pyramid Solitaire – *The best just got better!* V3.0 of this popular game is now available.

CoCothello – The popular board game adapted for your Color Computer 3 and OS9 Level 2.

CoCoYahtzee – The well known "Dice Poker" game.
(requires 80 column monitor)

KnightsBridge – *BRAND NEW!* It's you against the CoCo, with an element of chance thrown in!

All game programs just ***$10*** each or any three for ***$25***. I'll pay the shipping! All programs require at least 256K except CoCothello which will run in 128K. All are completely Mouse Controlled and require the WindInt module.

Send check or Money Order to:
ColorSystems
P.O. Box 540
Castle Hayne, NC 28429–0540

(NC residents please include 5% sales tax.)

ADVERTISING INFORMATION

Color Computing accepts display and Classified advertisements. When display advertising, you have an option of placing 1/4, 1/2, and full page advertisements. You can also place an advertisement on the inside front or back cover of the magazine. To receive all our rates and more detailed information, please request our **Advertising Rates and Information** packet.
(Classified advertising rates are $.60 per line, with a minimum of 3 lines.)

PROGRAM SUBMISSIONS

Attention all programmers! If you have a BASIC/OS-9 program that you have written, and would like to submit it to *Color Computing,* please send it to us on tape or disk. (We're sorry but we can't type in listings.) If the program meets our needs for a particular issue, we will publish it. You will be notified before your submission is published. Make sure the program is error-free, and any instructions or descriptions are included with it. We are currently looking for 1 or 2 line programs (which may be typed on a regular piece of paper), and OS-9 programs for **The Shell.**

* For more information on submitting material to *Color Computing* such as articles and product reviews, ask for our **Submission Information** packet and the **1992 Editorial Calendar.**

Note: All submissions become the property of Color Computing magazine.

DO YOU HAVE ANY QUESTIONS REGARDING THIS MAGAZINE? DIRECT ALL INQUERIES TO OUR ADDRESS ON THE BACK COVER, OR FEEL FREE TO GIVE US A CALL AT (617) 828-7749. WE'D BE GLAD TO HEAR FROM YOU!

ADVERTISER'S INDEX

NOTE: TC stands for TRS-80 Computing.

COLOR COMPUTING
65 OAK ROAD
CANTON, MA 02021-2605