# DYNAMIC COLOR NEWS

## Radio Shack Color Computer
## Magazine

BASIC PROGRAMMING

ASSEMBLY LANGUAGE

**OUR LAST ISSUE**

OS-9 & BASIC 09

PROGRAMS

HAM RADIO

C LANGUAGE

## RAMDISK
### for the 512K COCO 3

A ramdisk operates similar to a disk drive except it is many times faster. The 512K ramdisk allows drives 2 and 3 to be ramdisks. You can backup a disk to either ramdisk or select either one for quick program or data loading. OS-9 is not required. A memory test program is also included. $15

## DYPRINT

Now you can print LARGE signs for special occassions such as birthdays, parties, or yard sales. Even make your own FOR SALE signs when you need to sell that old car or lawnmower. BANNER uses standard print characters and is compatible with any printer. The characters are formed by a 21 x 27 dot pattern and are printed sideways across the paper. The basic character can be expanded up to 4 times for making large characters up to a full page.

MAXPRINT allows graphics to be blown up and printed on a standard printer. Any PMODE 4 picture can be printed. The program supports all 8 graphics pages for a total of 12288 bytes. MAXPRINT prints 8 characters per byte for a total of 98304 characters. Blow up pictures of friends and family generated by the DS-69B digitizer or make posters announcing sales or special events.

The DYPRINT package contains both BANNER and MAXPRINT. The cost is only $19.95

## NEW TERMINAL PROGRAM

DYTERM 2 - Allows a Color Computer to interface with Modems, Terminals, or other Computers using the ASCII port. 300-2400 baud, 1 or 2 Stop bits, 7 or 8 bit words, variable parity. Download programs from bulletin boards or other computers or upload your ASCII programs. Supports CoCo 2 and CoCo 3 Disk or Tape computers. Basic program with machine language subroutines is easily modified.
Tape or Disk $19.95.

## DECIMAL ML ASSEMBLER

DISASM is a 6809 Assembler-Disassembler that allows machine codes to be assembled using English mnemonics & decimal arithmetic. It supports all 6809 codes and is especially useful for beginners. Learn Assembly programming without using hex. Disassemble machine language programs and print them to a printer. $9.95

## DS-69B DIGITIZER

Capture pictures from your VCR or video camera. Then print them on your graphics printer. Have your friends over for an evening of fun and digitize and print their pictures. Supports all color computers. The picture can be displayed on the COCO 3's high resolution screen. Pictures can be Labeled with COCO MAX and printed on a graphics printer or saved on disk. 256 x 256 resolution, 64 levels of grey, & 8 images per second. Plug in ROM pack requires a multipack expander. Works with all color computer disk systems.
DS-69B $149.95 including shipping.

## CC-THERM 2

CC-THERM 2 is a dual digital thermometer for Radio Shack Color Computers. It consists of two thermistors wired to the end of 10' and 20' flat cables for measuring inside and outside temperatures. The other end of the cable is wired to a joystick plug. The thermistors can be mounted on a wall, inside equipment, or outside for temperature measurements. Basic software on tape or disk continuously prints the temperature in both Fahrenheit and Centigrade. T or D software. $19.95

## CC-LT

Now you can measure both temperature and light. The joystick assembly includes a light and temperature sensor at the end of a 20' flat cable. Uses one joystick plug. T or D Software 19.95.

## MEMORY MANAGER
### (for the Color Computer 2)

Did you know that the 64K Color Computer 2 and earlier computers have an extra 32K that is generally not used? Our Memory Manager allows basic or machine language programs to be run in either 32K bank. Banks are exchanged with an EXEC command. Also the second bank can be used as a ramdisk to store programs. This makes cassette operation faster than a disk. A third option configures the computer for the all ram mode allowing data or programs to be stored in the upper memory. The Memory Manager software is available on either cassette or disk. $19.95.

## MEMORY SAVER 2

Have you ever had a power failure or brownout to wipe out your program? The Memory Saver II is a battery backup assembly that prevents loss of programs due to power failures. It mounts under the keyboard and works with all color computers. Consists of gel rechargeable battery, control circuit, & miniature toggle switch. Will power a color computer for up to a couple of hours during a power failure.
Special sale price. $29.95.

The purpose of this magazine is to provide instruction on Basic & Machine Language programming, Computer theory, operating techniques, computer expansion, plus provide answers to questions from our subscribers.
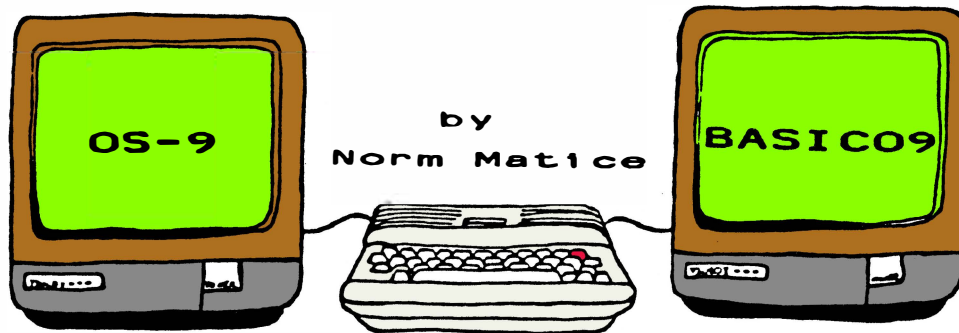
The submission of questions, operating hints, and solutions to problems to be published in this magazine are encouraged. All submissions become the property of Dynamic Electronics, Inc. if the material is used. We reserve the right to edit all material used and not to use material which we determine is unsuited for publication.

We encourage the submission of Basic and Machine Language Programs as well as articles. All Programs must be well documented so the readers can understand how the program works. We will pay for programs and articles based upon their value to the magazine. Material sent will not be returned unless return postage is included. Basic & ML programs should be sent on a tape or disk & comments should be sent as a DAT or TXT file.

## CONTENTS

* Included on Disk or Tape.
# OS-9 and C Procedures are included on DCN on DISK.

OS-9   by Norm Matice   BASIC09

## *MAKING XMODE CHANGES PERNAMENT*

Last month I promised to tell you how to make XMODE changes permanent. This will be accomplished by using the COBBLER command supplied with OS-9.

Before we get into the exact proceedure to follow, let's review a few safety precautions first.   Because we are making a permanent change to the system disk, we had better make sure we are using a copy of our system master. That way if something were to go wrong, such as a typing error in entering our command, we will still have a way to restore our system disk.

For our demonstration, let's change the baud rate on the printer. To find our current baud rate we will type in the following command at the OS-9 prompt.

XMODE /P

This will give us a list of parameters for the printer. Near the end of that list is the printer's baud rate.   If your system has never been changed the baud rate should equal 02. This is equivalent to 600 baud. We will change our system to 9600 baud. If you want to change to one of the other baud rates available,  check out page 6-103 in the system commands description part of your OS-9 manual. To change to 9600 we will need to  use the XMODE command, as we did last month. Type in the following line.

XMODE /P BAUD=06

This will effect the change

for our system.  Now if we were to reboot the system the baud rate would once again be 600. To insure that the baud rate is 9600 upon booting the system, we would follow the preceeding command above with the following command. When the OS-9 prompt reappears after the above change type:

COBBLER /DO

This makes the change permanent. Now whenever we boot the system the printers baud rate will be 9600. To check this out push the reset button on the back of your computer.  After the system has rebooted type in the following line:

XMODE /P

You should see baud=06 near the end of the information.   If you look at the book, on page 6-103, you will see that that value corresponds to 9600 baud. To change back to 600 baud or some other value use the XMODE command again and follow it again with the COBBLER command.

If your printer doesn't respond too good to the new baud rate, we can make changes to the baud rate with the TUNEPORT command. To find out what your current baud rate is type the following command in at the OS-9 prompt.

TUNEPORT /P

It will show your current baud rate and send a sample to

the printer. Make sure your printer is on when you use the command. After the sample print it will ask for a new rate. You may enter what you believe to be an appropriate rate and the system will send another sample to the printer. From this you can judge how your change will work. Once you have your system tuned the way you want it, it will stay that way as long as the system is booted up. Upon the next boot the system will go back to what it was. To make your change permanent you will need to use the COBBLER command again. Use it just as we did for the previous example.

We have now covered most of the modules in OS-9. Let's see if we can run down a few that have escaped us so far. There is a command to check the disk's file structure. It is the DCHECK command. To get a list of DCHECK's options type in the following line:

```
DCHECK -O
```

This should result in a list of options being listed on your screen. At the bottom of the option list is the option to give a count of the number of files and directories on your disk. Let's use that to demonstrate the syntax for this command. We will need to type in the command, the option it is using if any, and the disk drive we wish to check. Type in the following line:

```
DCHECK -S /DO
```

I would expect everyone's count to differ by a bit. On my system I came up with 4 directories and 75 files on the system disk I was using. Next month we will try to round up the few stray commands still wandering around.

## BASIC09

This month we will explore the debug feature of BASIC09. Through the use of some examples we will get the chance to use some of the debug command. Now would be a good time to fire up your machine and get BASIC09 up and running.

Type or load in the first listing below. As you can see it is a short program that doesn't accomplish much. One of the things it does do is ask the computer to divide by zero. When the value of A hits 0 in the loop the computer will dump us into the debug mode.

```
PROCEDURE DIVZERO
DIM A,B,C:INTEGER
A=25
WHILE A>-20 DO
B=50/A
PRINT B
A=A-1
ENDWHILE
```

When we see the D: prompt we know that we are indeed in the debug mode. In this case the computer shows us the offending line and gives us an error code. Error #045 according to page A-2 in our BASIC09 manual is divide by zero. Being a little dubious, let's check this out. In the debug mode we can check the current value of a variable just by issuing the correct command. Type PRINT A. Sure enough we were trying to divide by zero.

Now that we are in the debug mode how do we get out? If we type a Q at the D: prompt we will be returned to the B: prompt of BASIC09. What if we would like to get into debug from BASIC09? Type the RUN for your procedure at the B: prompt. Before our program reaches the point where it divides by zero, hold down the CTRL key and press the C key.

Now that we are in the debug mode what will we do?m Since we know the program will bomb when

the value of A reaches 0 let's change the value of A.

We can modify A's value to -1, the point it would reach right after the error. Type LET A := -1. Now that the value of A is changed let's see if the program will run error free the rest of the way. To get program execution to continue type CONT at the D: prompt. It would appear that we only have the one error in our program.

While we are in the debug mode we can also look at the BASIC09 directory by tying DIR at the D: prompt. You can also execute OS-9 commands from debug. Type a $ followed by the command. For example, at the D: prompt, type $ DIR and you will get the directory of your disk. Also available to you in debug is the ability to list your current procedure. Type LIST at the D: prompt.

Now enter listing two into your computer. This procedure figures out the sine of 90 degrees and then drops into the loop. Upon exiting the loop it calculates the sine of PI/2. If you run the program you might notice that the sine of 90 degrees is 1, the sine of PI/2 radians should also be 1 but the computer records it as 2.74121336E-02. Run the program again, but this time enter the debug mode with CTRL C while we are in the loop. Type list at the debug prompt. Notice how DEG was used to specify that our figures are in degrees. This makes us suspect perhaps PI/2 radians and 90 degrees are two different formats of the same number.

```
PROCEDURE CONVER
DIM A,B:REAL
DIM X:INTEGER
DEG
A=90
B=SIN(A)
PRINT B
PRINT "ENTERING LOOP"
```

```
FOR X=1 TO 10000
NEXT X
PRINT "OUT OF LOOP"
A=PI/2
B=SIN(A)
PRINT B
```

Let's change the format while we are in debug. Type in RAD and press enter then type CONT and press enter. Sure enough the program now shows that the sine of 90 degrees and PI/2 are both 1, as we knew they should be.

Last on the agenda is the TRON/TROFF functions. Type in the third example program. When you run the program the PAUSE that follows the DIM statement will put you into the debug mode. At the D: prompt type in TRON and press enter. Now type in CONT and press enter. The procedure will continue to run with the results of each step shown on the screen. TROFF of course will turn this feature off when it is used. If the procedure is done running it will automatically shut off. The debug mode is covered in chapter 5 of your BASIC09 reference manual.

```
PROCEDURE INC
DIM A,B,D,E,F,G:INTEGER
PAUSE
A=5
B=10
C=2
D=20
E=A+B
F=E*C
G=F-D
PRINT G
```

# TAXMAN

## game

You are a tax collector and are trying to accumulate as many dollars as possible. However there is a homeowner who is trying to destroy you before you collect your taxes. Use the arrow keys to move around the maze and finish as many levels as you can. This is an exciting one player game and is provided as a courtsey of T & D Subscription Software (See their advertisement on page 9).

```
1 REM COPYRIGHT (C) T&D SOFTWARE
  1987          taxman

10 CLS:PRINT@44,~taxman~;:PRINT@
97,~THE UNLUCKY HOMEOWNER WILL N
OT~;:PRINT@129,~PAY HIS TAXES.YO
UR JOB IS TO~;:PRINT@161,~COLLEC
T ALL THE MONEY THE OWNER~;:PRIN
T@193,~HAS.WATCH OUT FOR THE OWN
ER,~;:PRINT@225,~BECAUSE HE IS M
EAN AND DOES NOT~;

20 PRINT@257,~LIKE YOU AT ALL.~;
:PRINT@289,~USE ARROW KEYS TO MO
VE AROUND~;:PRINT@321,~THE MAZE.
~;:PRINT@483,~PRESS <enter> TO C
ONTINUE~;
30 IFINKEY$<>CHR$(13)THEN30

40 CLS:PRINT@44,~taxman~;:POKE11
25,36:PRINT@103,~= MONEY~;:PRINT
@165,~o = HOMEOWNER~;:PRINT@229,
~x = TAX COLLECTOR(YOU)~;:PRINT@
325,~YOU HAVE FOUR LIVES AT~;:PR
INT@357,~ THE BEGINNING OF THE~;
:PRINT@398,~GAME~;:PRINT@485,~PR
ESS <enter> TO START~;
50 IFINKEY$<>CHR$(13)THEN50
60 CLS:M=175:D=36:LV=1
65 FORI=0TO63:POKE1024+I,32:NEXT

70 RESTORE:FORI=64TO510:READA:IF
A=1THENPRINT@I,CHR$(175);ELSEIFA
=.THENPOKE1024+I,36:ELSEIFA=2THE
NPOKE1024+I,32

80 NEXT:P=335:POKE1535,175:IFLV>
```

```
10THENLV=10

90 PRINT@1,~score~;:PRINT@13,~ta
xman~;:PRINT@24,~level~LV;

100 PRINT@32,SC;:A$=INKEY$:IFA$=
~BTHENY=-32ELSEIFA$=CHR$(10)THE
NY=32ELSEIFA$=CHR$(8)THENY=-1ELS
EIFA$=CHR$(9)THENY=1

110 IFP=289THENP=318:POKE1313,32
ELSEIFP=318THENP=289:POKE1342,32
ELSEIFDT=1THENV=-1

120 B=PEEK(P+Y+1024):IFB=175THEN
Y=0ELSEIFB=36THENDT=DT+1:SC=SC+(
LV*5)ELSEIFB=35THENPRINT@238,LV*
50;:FORI=1TO10:SOUND200,1:NEXT:P
OKE1262,32:POKE1263,32:POKE1264,
32:POKE1265,32:POKE1266,32:SC=SC
+LV*50

130 POKE1024+P,32:P=P+Y:PRINT@P,
~x~;:P2=POS(0)

140 PRINT@M,~o~;:M2=POS(0):P3=IN
T(P/32):M3=INT(M/32)

150 IF P3>M3 THENV=32ELSEIF P3<M
3 THENV=-32

160 IFPEEK(1024+M+V)=175 ANDP2>M
2 THENV=1

170 IFPEEK(1024+M+V)=175 ANDP2<M
2 THENV=-1

180 IFDT=199THENLV=LV+1:RESTORE:
P=335:DT=0:D=36:GOTO380

190 IF PEEK(1024+M+V)=175THENV=0
ELSEPOKE1024+M,D:M=M+V:D=PEEK(10
24+M)

200 W=RND(200):IFW=1THENPOKE1359
,35

210 E=RND(1000)/LV:IFE=1THENPOKE
1359,32
220 IFD=24THEN390
230 GOTO100
```

```
240 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1

250 DATA1,,,,,,,,,,,,,,,1,1,,,,,
,,,,,,,,,1

260 DATA 1,,1,,1,1,1,1,1,1,1,1,1
,1,,1,1,,1,1,1,1,1,1,1,1,1,,1,
,1

270 DATA 1,,,,,,,,,,,,,,,,,,
,,,,,,,,,1

280 DATA 1,,1,,1,1,1,1,1,1,1,,1,
1,1,1,1,1,1,,1,1,1,1,1,1,1,,1,
,1

290 DATA1,,,,,,,1,,,,,1,2,2,2,2,
2,2,1,,,,,1,,,,,,,1

300 DATA1,1,1,,1,1,,1,,1,1,,1,2,
2,2,2,2,2,1,,1,1,,1,,1,1,,1,1,1

310 DATA1,2,2,,1,1,,1,,1,1,,1,1,
1,1,1,1,1,,1,1,,1,,1,1,,2,2,1

320 DATA1,1,1,,,,,,,,,,,,,,,,,,,
,,,,,,1,1,1

330 DATA1,,,,1,1,1,1,1,1,1,1,1,1
```

```
,1,1,1,1,1,1,1,1,1,1,1,1,1,,,,
1

340 DATA1,,1,,,,,,,,,,,,,1,1,,,,
,,,,,,,1,,1

350 DATA1,,1,,1,1,1,1,1,1,1,1,1,
1,,1,1,,1,1,1,1,1,1,1,1,1,1,,1,,
1

360 DATA1,,,,,,,,,,,,,,,,,,,,
,,,,,,,,1

370 DATA1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1

380 Y=0:POKE357,59:FORI=1TO5:SCR
EENO,1:FORT=1TO500:NEXTT:SCREENO
:FORW=1TO500:NEXTW,I:P=335:M=175
:GOTO70

390 PLAY"T100;O1;CDACBDCBFGDCCBD
GACEFDGCBDGACECBADCBEDC":POKE102
4+P,32:N=PEEK(1199):D=N:MN=MN-1:
IFMN=-4THEN400ELSEP=335:M=175:GO
TO100

400 PRINT@238,"game";:PRINT@270,
"over";

410 SCREENO,1:SCREENO:GOTO410
```

# ASSEMBLY LANGUAGE

by
Doug Canfield

This month I will explain more of the various modes of addressing. Exactly what do we mean by "addressing"? When the computer has to perform a computation, it must know where to go to get its data. There are several options. It might be stored in the program itself, in a known memory location, or we might have to look at a specified memory location just to find out where it is stored. Each of these different ways of getting data is known as a different form of addressing. The first, immediate addressing, we covered last month, and so we will deal with the others this month.

## Extended Addressing

In our program last month, we stored the word "NEW " in the first four positions of the screen. Suppose that we wanted to copy these four letters to a different location on the screen. We could just rewite the routine to first put these letters at the beginning, and then store them again somewhere else, but there is an easier way...

Instead of telling the computer exactly what letters we want to put on the screen we will tell it to look at the first five positions of screen memory to find them. By doing this, we won't be restricted to the word "NEW ", but we could move "OLD ", "BARN", "GIFT", or any other word under four letters long.

When the microprocessor uses information in a predetermined place in memory it is called "exteneded addressing". We use this form of addressing when we know where the information that we are using is stored, but we don't have to know what it is.

A short program to move four characters on the screen will illustrate extended addressing.

```
7000 B6    0400    LDA $0400
7003 B7    0500    STA $0500
7006 B6    0401    LDA $0401
7009 B7    0501    STA $0501
700C B6    0402    LDA $0402
700F B7    0502    STA $0502
7012 B6    0403    LDA $0403
7015 B7    0503    STA $0503
7018 39            RTS
```

In last month's program, the first number that the 6809 encountered was "$86"; in this month's program you can see that it is a "$B6". Whereas "$86" tells the microprocessor to get the next number in the program, and put IT in the A register, "$B6" has it look at the next TWO numbers in the program, to find the memory ADDRESS where the number is stored that will be put into the A register.

NOTE: A $ preceeding a number means that it is a hexadecimal number. Most assemblers will take decimal numbers, in which case the $ symbol will be eliminated. - Editor.

This month's program steers the computer to $400-$403 to get the numbers that it puts at $500-$503. You may have noticed that the instruction at $7003 (B7) has not changed from last month. An immediate addressed store is senseless (storing a number in to the next program location), and doesn't exist as such. Both this and last

month's stores were all in extended addressing.

## Indexed Addressing

By far the most powerful mode of addressing is indexed addressing. In and of itself it has a multitude of variants, which I will cover in sequence.

-When the microprocessor uses its own internal registers to find the address in memory with which to transfer data, it is called indexed addressing. We use this form of addressing when we want to work with a "block" of data in memory.

As in extended addressing, indexed addressing transfers data between the microprocessor and another place in memory. It differs; however, in that instead of the microprocessor looking to the next numbers in the program to find the memory address, it looks to a number stored in one of its internal registers. In order to get the proper data, the programmer or the program must put the correct data into this register. The microprocessor can use five of its 16 bit registers (X, Y, S, U, and PC) to find the memory address it needs. If $400 is stored in the X register, and we use a LDA instruction indexed (pointing) to the X register, the microprocessor will load its A register with the number in memory address $400 (the first position on the text screen). Were this all that we could do with indexed addressing, it would be of very limited value; but, as we examine some of the many variants, we will begin to see some of the power of our microprocessor shine through. Power, incidently, which isn't available to many other microprocessors.

If we are using only one piece of data, we normally use extended or immediate addressing. If; however, we must use many pieces of data, stored in a block, then we would use indexed addressing. The microprocessor can add a value to the one stored in the "index" register (X, Y, S, U, or PC) to locate many memory addresses other than one pointed to by the index register. We will now look at some ways it can accomplish this.

## Offset Indexed

We can have the microprocessor internally add an offset value to the one stored in the index register to calculate the "effective" address that it will use to store or retreive data. The program below uses this technique to show another way of moving our letters around on the screen.

```
7000 8E    0400   LDX #$0400
7003 108E  0500   LDY #$0500
7007 A6    84     LDA ,X
7009 A7    A4     STA ,Y
700B A6    01     LDA 1,X
700D A7    21     STA 1,Y
700F A6    02     LDA 2,X
7011 A7    22     STA 2,Y
7013 A6    03     LDA 3,X
7015 A7    23     STA 3,Y
7017 39           RTS
```

The program above uses the X register to point to the beginning of the characters we want to move, and the Y register to point to the beginning of where we want to move them. I will first explain the right two columns. The # sign means immediate addressing, and the comma (,) indicates some form of indexed addressing. The rest are extended addressing. As for the indexed addressing, the letter to the right of the comma is the register to use, and the number to the left of the comma is the value that the microprocessor will add to the register to form its effective address.

11

This can be any number from -32768 to 32767 (decimal). Writing a zero to the left of the comma is optional.

(Note that this calculation is done entirely internally; the number in the index register is not changed.)

Looking at columns 2&3 (the actual hexadecimal numbers in memory) will show some new insights. The first thing to notice is the series 108E starting at $7003. Most of the 6809's instructions are one byte long, but as one byte limits you to 256 instructions, two bytes had to be used to get more. All of the two byte instructions begin with $10 or $11, and their second byte is always the same as an instruction that does the same thing with a different register, or in a slightly different way. The instruction set for the 6809 is "optimized"; that is, all of the most common instructions use one byte.

Beginning at 7007, you can see that the code for LDA, in this case is $A7. The microprocessor interprets this as an indexed load into the A register, and looks to the next several bytes to see what index register to use, and how much to add (or subtract) from this index register to form the effective address. I will explain how to interpret these "post-bytes" next month, but for now, suffice it to know that $84 means "Use the X register, and don't add any offset." The rest of the program is fairly self explanatory. The advantages of this program are many-fold.

1) Its shorter.
2) Its faster.
3) Its easy to change the source and destination of where to move the characters.

## ACCUMULATOR OFFSET

The last type of indexed addressing that we will cover this month is accumulator offset. In this mode, the microprocessor doesn't add a constant offset to the index register, as in the preceeding example, but it adds one of the accumulators (A, B, or D) to the index register to get the effective address. To get the most benefit from this type of addressing, we must use some concepts that we haven't covered yet (branching, looping, and incrementing), but I plan to cover some of these next month.

# CHARACTER GENERATOR

## for the Color Computer II

One of the drawbacks of the color computer 2 and earlier computers was the 32 character screen. It is possible to use the graphics screen and generate a different character set. This program from Bill Bernico Software allows 51 characters across and 24 lines which allows much more information to be displayed than could be presented with the standard 32 character by 16 line display. The program may not work with all graphics programs and it also slows the computer down a little. However the characters are well defined and the program can be used for many applications.

A machine language program is generated and placed in upper memory. It takes a few seconds for the program to generate the machine language program. After it is completed, the computer can be used as normal with the expanded character screen.

```
1 '51X24 TEXT MODE FROM BASIC
   (C)1988 FROM
   BILL BERNICO SOFTWARE

2 'THIS SUBROUTINE WILL PUT YOUR
   COMPUTER INTO A 51 X 24 TEXT
   MODE.

3 IF PEEK(31847)=142 THEN3ELSECL
EAR1,31846

4 DATA142,124,253,252,1,104,191,
1,104,253,125,144,142,124,134,22
0,169,159,169,253,124,157,158,18
6,191,127,252,127,127,250,57,129
,158,38,21,52,22,158,186,191,127
;252,204,255,255,237,129,156,183
,38,250,53,22,126,16,0,129,135,3
8,5,115,127,255,32,244,12

5 DATA64,39,9,19,32,39,236,127,
127,255,32,231,125,127,255,39,22
6,127,127,255,52,22,134,32,23,0,
```

```
241,189,179,228,16,131,4,200,16,
36,55,123,16,131,0,50,35,8,131,0
,51,124,127,255,32,242,134,5,61,
52,4,196,7,247,127,250,182,127,2
55,53,4,84,84,84,211,186

6 DATA253,127,252,127,127,255,53
,22,134,32,32,159,13,111,16,38,0
,140,52,22,129,31,38,15,134,240,
177,127,249,38,4,141,94,32,134,1
41,92,32,130,129,8,38,21,23,1,44
,134,32,23,0,145,134,32,23,0,140
,23,1,31,23,1,28,32,54,129,12,38
,19,158,186,191,127,252

7 DATA127,127,250,204,255,255,23
7,129,156,183,38,250,32,31,141,7
2,190,127,252,156,183,37,22,48,1
37,255,0,191,127,252,158,186,236
,137,1,0,237,129,188,127,252,38,
245,32,214,134,95,141,74,23,0,22
1,53,150,134,15,183,127,249,198,
94,142,126,115,48,3

8 DATA182,127,249,129,15,38,4,17
0,132,32,2,164,132,167,128,90,38
,236,57,126,16,0,129,13,38,19,14
1,216,134,32,141,25,23,0,172,127
,127,253,124,127,252,127,127,250
,57,129,32,16,37,254,237,129,128
,16,34,254,231,198,4,247,127,254
,128,32,61,195,126

9 DATA115,31,2,190,127,252,166,1
60,52,2,138,15,141,22,53,2,72,72
,72,72,138,15,141,12,122,127,254
,38,233,48,137,255,0,141,66,57,2
46,127,250,92,247,127,251,52,2,2
04,248,0,138,8,253,127,247,53,2,
198,255,122,127,251,39,14,26,1,7
0,86,28,254,118,127

10 DATA247,118,127,248,32,237,52
,6,252,127,247,170,132,234,1,167
,132,231,1,53,6,164,132,228,1,23
7,132,48,136,32,57,246,127,2500,2
03,5,193,7,35,4,48,1,192,8,193,7
,38,15,30,16,193,31,38,7,76,95,3
0,1,95,32,2,30,1,247,127,250,191
,127,252,57,52,4,246

11 DATA127,250,190,127,252,192,5
```

```
,42,20,203,8,48,31,30,16,193,255
,38,8,198,31,30,16,198,2,32,2,30
,16,191,127,252,247,127,250,53,1
32,255,255,255,255,221,221,223,2
23,85,255,255,255,153,9,9,159,17
7,123,209,191,243,45,180,207,181
,91,37,175,219,255,255

12 DATA255,219,119,123,223,189,2
38,237,191,246,144,150,255,251,1
77,187,255,255,255,157,191,255,2
40,255,255,255,255,253,223,255,2
37,191,255,150,64,38,159,217,221
,221,143,150,237,183,15,150,233,
230,159,217,80,221,223,7,30,230,
159,219,113,102,159,14

13 DATA237,183,127,150,105,102,1
59,150,104,237,191,253,223,221,2
55,253,223,221,191,237,183,189,2
39,255,15,15,255,123,222,219,127
,150,237,191,191,150,66,71,143,1
50,96,102,111,53,81,102,31,150,1
19,118,159,26,170,170,31,7,113,1
19,15,7,113,119,127

14 DATA150,116,102,159,102,96,10
2,111,141,221,221,143,206,238,23
0,159,101,51,53,111,119,119,119,
15,96,6,102,111,98,36,68,111,150
,102,102,159,22,97,119,127,150,1
02,37,175,22,97,53,111,150,121,2
```

```
30,159,13,221,221,223,102,102,10
2,159,102,102,105,159

15 DATA102,102,0,111,102,153,150
,111,102,96,221,223,14,201,55,15
,139,187,187,143,255,251,222,255
,29,221,221,31,253,141,221,255,2
55,255,255,15,189,255,255,255,25
5,30,134,143,119,113,102,31,255,
241,119,31,238,232,102,143,255,1
50,7,157,159,218,177,187

16 DATA191,255,6,14,143,119,22,1
02,111,223,157,221,143,239,238,2
30,159,247,100,22,111,59,187,187
,31,255,96,102,111,255,22,102,11
1,255,150,102,159,255,6,7,127,25
5,134,174,239,255,22,119,127,255
,7,14,15,187,27,187,191,255,102,
102,159,255,102,105

17 DATA159,255,102,96,111,255,10
5,150,111,255,102,157,159,255,14
,219,15,219,179,187,223,189,220,
221,191,255,10,170,175,170,170,1
70,170,85,85,85,85,0,0,0,0,83,84
,65,82,84,124,103,0,0,69,88,142,
128,222,206,1

18 FORX=31847TO32767:READX,Y:POKEX
,Y:NEXT:PMODE4,1:PCLS1:SCREEN1,1
:EXEC31847:CLEAR200
```

# BASIC PROGRAMMING

In this series I am showing how to use basic commands to write effective and useful programs. The version of basic that Radio Shack Color Computers use, is the best version of basic which was developed by Microsoft that I have seen. Most of the commands are usable on other computers. For example I have a Radio Shack model 100 portable computer. I can write programs on it that will also work on my color computers. I also have IBM compatible computers which use GW basic. This version of basic is very similar to Radio Shack's Extended Color Basic. Most of the commands have been covered, and for the past few months I have been concentrating on giving example programs.

USIari-
ables that can be used in basic programming. Numerical variables represent numbers which can be used in calculations. String variables allow us to manipulate word phrases. We use words in everything we do. When we write a basic program, we use words to represent our commands. These words have to be translated into commands for the computer to use. This is done by basic each time we enter a program line.

One powerful feature of basic is its housekeeping ability. The term housekeeping means to keep things organized. When we define a variable such as X or X$, basic has to put the values of the variables somewhere in memory and keeps up with where it is located.

Last month I presented a program that read in a string from a disk or cassette, checked it to see if it was in the proper format for basic, and printed the string to another disk or cassette file. If a string were not suitable for basic, then it was not printed to the file. This allowed a basic program, which was saved in ASCII, to be corrected so that it could be loaded. It could then be edited to replace the lines that were removed due to errors. Saving programs in ASCII format allows them to be recovered if errors creep in. A word processor can be used to edit them.

WORD PROCESSING WITH STRINGS

Let's develop a simple but useful word processor program using strings. Our word processor will be compatible with other word processors and will work with both a cassette and disk. First we will have to establish an array for the strings by using a dimension statement similar to the following:

10 DIM X$(400)

If we used the preceeding statement in our program, then we could have a maximum of 400 lines in our word processor file. An array simplifies programming by allowing us to use FOR-NEXT loops for inputting, saving, and printing the file. With an array we can easily eliminate a line or insert a line or lines. Our word processor program will be relatively short but it will need to allow maximum memory for the strings. This can be accomplished with the CLEAR and PCLEAR commands.

The PCLEAR command determines the number of graphics pages that are reserved in memory below the start of basic. The computer initializes with 4 graphics pages reserved. The PCLEAR 4 command will give this startup situation. However the PCLEAR 1 command allows for only one graphics page and more memory for strings. Since our program will not use graphics, the PCLEAR1 command is what we want.

The CLEAR command reserves memory for strings. If we are to have 400 lines or strings, then a lot of memory will be required. The number after the CLEAR command is the amount of memory that is reserved for strings. This should be a minimum of 15000 bytes. So we would use CLEAR 15000 to reserve 15000 bytes of memory.

### THE MENU

When I develop a program, I first decide the features I want the program to have. As I list these, I generate a menu. Then I give a prompt to select one of the features of the menu. I then branch to the part of the program that will accomplish the task I selected. To keep things straight I use numbers like 1000, 2000, 3000, 4000 , etc. to contain the various parts of the program. I then write each section as a sub program.

### STRING$ COMMAND

A very useful command that I used in the word processor program is the string$ command. It generates a string of length L of any character. For the left margin I used the variable MM to stand for the number of characters in the left margin. I needed a string of MM characters to print to the printer before I printed each line. So I used the following command:

```
4030 S$=STRING$(MM,32)
```

32 is the value of the space bar. Consider the following chart and note the size of S$ for the values of MM:

```
   MM    123456789

    2 S$="   "
    5 S$="      "
    9 S$="          "
```

A print command for the string X$(J) with a margin of S$ is as follows:

```
4070 PRINT #-2,S$;X$(J)
```

In the program I made comments where I thought they would be helpful. These can be edited out if you want to conserve space. The program allows ASCII files to be created, edited, loaded, saved, and printed. It is compatible with other word processors and allows basic programs to be written. The LINEINPUT command allows commas to be used within the string. The program is limited by the number of lines that it will handle, although 400 lines is quite a number. It could easily be modified to add additional features as required.

### WORD PROCESSOR PROGRAM

```
5 CLS
10 PRINT"WORD PROCESSOR PROGRAM
20 PRINT"USING STRINGS FOR DATA
30 PRINT"WRITTEN BY BILL CHAPPLE
35 PRINT"COPYRIGHT (C) 1989

37 PRINT"DYNAMIC ELECTRONICS INC

38 FOR J=1 TO 600: NEXT

40 PCLEAR1:CLEAR 15000 'CLEAR SP
ACE FOR THE STRINGS

50 DIM X$(400) 'SET UP THE ARRAY
S FOR 400 LINES
55 'THIS IS THE MENU OF OPTIONS
60 CLS:PRINT"1 LOAD FILE
70 PRINT"2 SAVE FILE
80 PRINT"3 REVIEW & EDIT FILE
90 PRINT"4 PRINT THE FILE
```

```
100 GOSUB 140 'THIS SUB GETS THE
 KEY PRESSED

110 ON X GOSUB 1000,2000,3000,40
00 'THIS IS THE EASY WAY TO BRAN
CH
120 GOTO60
130 '
140 P$=INKEY$:IF P$="" THEN 140
150 X=VAL(P$):RETURN

160 PRINT"ENTER FILE NAME/EXT":I
NPUT N$:RETURN
170 '
1000 PRINT"THIS LOADS A FILE"
1010 PRINT"D-DISK FILE
1020 PRINT"C-CASSETTE FILE

1030 GOSUB 140 'GO GET A FILE NA
ME

1040 IF P$="C" THEN 1050 ELSE IF
 P$="D" THEN 1080 ELSE 60

1050 PRINT"THIS LOADS A CASSETTE
 FILE. ":PRINT
1060 GOSUB160
1070 F=-1:GOTO1110

1080 PRINT"THIS LOADS A DISK FIL
E"
1090 DIR:GOSUB 160
1100 F=1:GOTO1110

1110 IF P$="C" THEN OPEN "I",#-1
,N$

1120 IF P$="D" THEN OPEN "I",#1,
N$ ELSE GOTO60
1130 FOR J=1 TO 400
1140 X$(J)="":NEXT
1150 FOR J=1 TO 400

1160 IF P$="C" THEN LINEINPUT #-
1,X$(J) ELSE LINEINPUT #1,X$(J)
1170 PRINTJ:PRINTX$(J)
1180 IF P$="D" THEN 1200
1190 IF EOF(-1)=-1 THEN 1220
1200 IF EOF(1)=-1 THEN 1220

1205 'THE PRECEEDING STATEMENTS
·CHECK FOR THE END OF THE FILE
1210 NEXT J

1220 XX=J:PRINT"THERE ARE "XX" F
ILES
1230 RETURN
1240 '
```

```
2000 PRINT"THIS SAVES A FILE

2010 INPUT"D FOR DISK OR C FOR C
ASSETTE";P$
2020 IF P$="D" THEN 2090
2025 'CASSETTE FILE
2030 GOSUB 160
2040 OPEN "O",#-1,N$
2050 FOR L=1 TO XX
2060 PRINT#-1,X$(L)
2070 NEXT

2080 CLOSE:PRINT"FILE IS SAVED O
N CASSETTE":GOTO2160

2086 'THE FOLLOWING IS FOR A DIS
K FILE
2090 CLS:DIR
2100 GOSUB 160
2110 OPEN "O", #1,N$
2120 FOR L=1 TO XX
2130 PRINT#1, X$(L)
2140 NEXT

2150 PRINT"THE FILE IS SAVED TO
DISK

2160 INPUT"PRESS ENTER TO CONTIN
UE";XP
2170 GO TO 60
2180 RETURN
2190 '

3000 CLS:PRINT"THIS REVIEWS AND
EDITS THE FILE
3010 PRINT"THERE ARE "XX" LINES
3015 'MENU FOR EDIT OPTIONS

3020 PRINT"L-ALLOWS ENTERING LIN
E NUMBER
3030 PRINT"D-DELETES A LINE
3040 PRINT"I-INSERTS A LINE
3050 PRINT"- MOVES BACK 5 LINES

3060 PRINT"SPACE BAR STARTS OR S
TOPS PGM

3070 PRINT"M RETURNS TO THIS MEN
U

3080 PRINT"E RETURNS TO MAIN MEN
U

3090 PRINT"ENTER LINE TO START O
R PRESS    ENTER FOR BEGINNING
3100 INPUTLN:IF LN=0 THEN LN=1
3110 FOR J=LN TO XX
```

```
3120 PRINT"
    "J

3130 PRINTX$(J) 'J=LINE NUMBER,
X$(J) IS THE LINE
3140 GOSUB 3420

3150 IF P$="E" THEN RETURN 'GO T
O MAIN MENU

3160 IF P$="M" THEN 3000 'GO TO
EDIT MENU
3170 FOR KK=1 TO 100:NEXT KK
3180 NEXT J

3190 PRINT"THIS IS THE END OF TH
E FILE.

3200 INPUT"PRESS ENTER";XM:GOTO6
0
3210 '

3220 PRINT"ENTER LINE TO INSERT,
 ENTER $ TO ABORT
3230 LINEINPUTL$
3240 IF L$="$" THEN RETURN

3250 INPUT"LINE NUMBER FOR THIS
LINE";LN

3260 XX=XX+1 'ADD 1 TO TOTAL NUM
BER OF LINES

3270 IF XX=400 THEN PRINT"TOO MA
NY LINES TO INSERT":RETURN 'MAXI
MUM OF 400 LINES

3275 'THE FOLLOWING MOVES ALL LI
NES DOWN ONE TO ALLOW A LINE TO
BE INSERTED
3280 FOR Y=XX TO LN+1 STEP -1
3290 X$(Y)=X$(Y-1)
3300 NEXT Y

3310 X$(LN)=L$ 'ADD THE NEW LINE
 TO THE ARRAY
3320 RETURN
3330 '

3340 INPUT"ENTER LINE NUMBER TO
DELETE";LN
3350 PRINTX$(LN)

3360 PRINT"PRESS Y TO DELETE THE
 LINE":GOSUB 140 IF P$<>Y THEN R
ETURN 'CHECK FOR Y

3365 'THIS MOVES ALL LINES UP ON
E
3370 FOR Y=LN TO XX
3380 X$(Y)=X$(Y+1)
3390 NEXT Y

3400 XX=XX-1:RETURN 'REDUCE TOTA
L NUMBER OF LINES BY ONE
3410 '

3415 'THE FOLLOWING DETECT WHICH
 KEY IS PRESSED AND CAUSES ACTIO
NS DEPENDING UPON WHICH KEY WAS
PRESSED.
3420 P$=INKEY$

3430 IF P$=" " THEN GOSUB 140 'S
PACE BAR

3440 IF P$="I" THEN GOTO 3220 'I
NSERT

3450 IF P$="L" THEN INPUT"ENTER
LINE NUMBER";J 'J IS LINE NUMBER

3460 IF P$="E" THEN GOTO3000 'EN
D THIS SECTION

3470 IF P$="D" THEN GO TO 3340 '
DISK

3480 IF P$="-" THEN J=J-5 'MOVE
UP FIVE LINES

3490 IF P$="M" THEN 60 'RETURN T
O MAIN MENU
3500 RETURN
3510 '
4000 PRINT"THIS PRINTS THE FILE

4010 INPUT"ENTER 1 FOR LINE NUMB
ERS";LL
4020 INPUT"ENTER LEFT MARGIN";MM

4030 S$=STRING$(MM,32) 'THIS GIV
ES A STRING OF MM SPACES FOR THE
 LEFT MARGIN
4040 FOR J=1 TO XX
4050 PRINTJ:PRINTX$(J)
4060 IF LL=1 THEN PRINT#-2,J;
4070 PRINT#-2,S$;X$(J)
4080 NEXT J
4090 RETURN
```
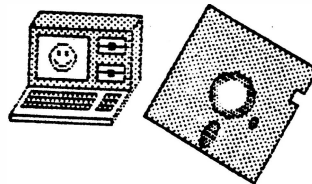
# PUBLIC DOMAIN SOFTWARE
## for Radio Shack Color Computers

*** PD-1 GAMES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| BEAST | BAS | 0 | B | 1 |
| BEAST | DAT | 1 | A | 1 |
| BOBO | BAS | 0 | B | 3 |
| GUNNER | BAS | 0 | B | 2 |
| HOW | BAS | 0 | B | 3 |
| LANDER | BAS | 0 | B | 3 |
| LIFE | BAS | 0 | B | 3 |
| MAX | BAS | 0 | B | 3 |
| POKER | BAS | 0 | B | 2 |
| BIORITHM | BAS | 0 | B | 3 |
| BLACKBOX | BAS | 0 | B | 2 |
| BLOCKADE | BAS | 0 | B | 1 |
| BUSJUMP | BAS | 0 | B | 1 |
| CHUTE | BAS | 0 | B | 2 |
| GO | BAS | 0 | B | 3 |
| HANGMAN | BAS | 0 | B | 2 |
| OTHELLO | BAS | 0 | B | 2 |
| TARTUS | BAS | 0 | B | 1 |
| TARTUS2 | BAS | 0 | B | 1 |

*** PD-2 GAMES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| RUBIC | BAS | 0 | B | 5 |
| FRACTAL | BAS | 0 | B | 1 |
| KALSCOPE | BAS | 0 | B | 2 |
| TARTUS | BAS | 0 | B | 1 |
| TARTUS2 | BAS | 0 | B | 1 |
| WORLD3D | BAS | 0 | B | 4 |
| LIFE | BAS | 0 | B | 2 |
| ADVENT | BAS | 0 | B | 4 |
| ADVENT | DOC | 1 | A | 2 |
| HURKLE | BAS | 0 | B | 2 |
| REVERSE | BAS | 0 | B | 2 |
| GUESSFR | BAS | 0 | B | 2 |
| SCRAMBLE | BAS | 0 | B | 3 |
| PIZZA | BAS | 0 | B | 2 |
| CINQUAIN | BAS | 0 | B | 2 |

*** PD-3 GAMES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| AANDAN | BAS | 0 | B | 2 |
| STARTREK | BAS | 0 | B | 9 |
| TREKINST | BAS | 0 | B | 3 |
| SEQUENCE | BAS | 0 | B | 2 |
| ALPHABET | BAS | 0 | B | 3 |
| GEOGRAPH | BAS | 0 | B | 4 |
| FLASH | BAS | 0 | B | 4 |
| BAGELS | BAS | 0 | B | 3 |
| OREGON | BAS | 0 | B | 9 |
| MULTIPLY | BAS | 0 | B | 2 |

*** PD-4 ML GAMES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| PONG | BIN | 2 | B | 1 |
| SQUASH | BIN | 2 | B | 2 |
| BLOCKADE | BIN | 2 | B | 2 |
| GERM | BIN | 2 | B | 1 |
| WIGWORM | BIN | 2 | B | 2 |
| GRID | BIN | 2 | B | 2 |
| ZEROG | BIN | 2 | B | 2 |
| 3DTICTAC | BIN | 2 | B | 7 |
| HOPBOP | BIN | 2 | B | 5 |
| ICEWAR | BAS | 0 | B | 6 |
| CIVILWAR | BAS | 0 | B | 4 |
| TICTACTO | BIN | 2 | B | 7 |

*** PD-5 GAMES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| CAVE | BAS | 0 | B | 4 |
| WARGAME | BAS | 0 | B | 2 |
| WARGAME | BIN | 2 | B | 1 |
| WARGAME2 | BAS | 0 | B | 5 |

| | | | | |
|---|---|---|---|---|
| WARROOM | BIN | 2 | B | 3 |
| NORAD | BAS | 0 | B | 3 |
| ANDREA | BAS | 0 | B | 5 |
| CURSE | BAS | 0 | B | 4 |
| GARGOYLE | BAS | 0 | B | 6 |
| KINGTUT | BAS | 0 | B | 7 |
| TAIPAN | BAS | 0 | B | 6 |

**DSK-6**

**SPELL & FIX- FIND
SPELLING ERRORS
IN TXT DISK FILES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| MANUAL | TXT | 1 | A | 12 |
| SPELLFX2 | BAS | 0 | B | 1 |
| SPELLFX2 | BIN | 2 | B | 6 |
| SPELLFIX | BAS | 0 | B | 1 |
| DICT | TXT | 1 | A | 33 |
| COREDICT | TXT | 1 | A | 1 |
| SAMPLE | TXT | 1 | A | 1 |
| BUILD | BAS | 0 | B | 1 |
| LIST | BAS | 0 | B | 1 |
| RESET | BAS | 0 | B | 1 |
| APPEND | BAS | 0 | B | 1 |
| ADDWORDS | BIN | 2 | B | 3 |

**PD-7 DISK UTILITIES**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| BASIC64 | BIN | 2 | B | 1 |
| BSEARCH | BIN | 2 | B | 1 |
| DISKCOMP | BIN | 2 | B | 1 |
| DISKTEST | BIN | 2 | B | 3 |
| DISKWASH | BAS | 0 | B | 1 |
| DOS64K | BAS | 0 | B | 2 |
| DSDBOOT | BIN | 2 | B | 1 |
| LIST | BIN | 2 | B | 2 |
| PRINT | BIN | 2 | B | 3 |
| PRINTDIR | BAS | 0 | B | 1 |
| RECOVER | BIN | 2 | B | 1 |
| ROMBACK | BAS | 0 | B | 1 |
| ROMFIX | BIN | 2 | B | 1 |

**PD-8 DISK UTILITIES**

| | | | | |
|---|---|---|---|---|
| SCRN51 | BAS | 0 | B | 1 |
| SCRN51 | BIN | 2 | B | 1 |
| SCRNDEMO | BAS | 0 | B | 2 |
| SDC | BIN | 2 | B | 1 |
| SQUEEZE | BIN | 2 | B | 1 |
| SSDBOOT | BIN | 2 | B | 1 |
| TAPE2DSK | BAS | 0 | B | 1 |
| TIMER | BIN | 2 | B | 2 |
| UNLOCK | BIN | 2 | B | 1 |
| BACKUP | BIN | 2 | B | 1 |
| BACKUP1 | BIN | 2 | B | 1 |
| MORE | BIN | 2 | B | 3 |
| SPEAK | BIN | 2 | B | 3 |
| PCLEARFX | BIN | 2 | B | 1 |
| MULTBACK | BIN | 2 | B | 1 |
| MULTBACK | DOC | 1 | A | 1 |

**PD-9**

**TERMINAL PROGRAMS**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| TELETERM | BIN | 2 | B | 3 |
| TELETERM | CAS | 2 | B | 3 |
| ITHELP | DAT | 1 | A | 4 |
| MTERM | BIN | 2 | B | 6 |
| MTERM1 | VIP | 1 | A | 3 |
| MTCONFIG | BAS | 0 | B | 3 |
| MTERM+ | BIN | 2 | B | 6 |
| DATATRDE | BIN | 2 | B | 3 |
| KERMIT | BAS | 1 | A | 1 |
| KERMIT | BIN | 2 | B | 2 |

| | | | | |
|---|---|---|---|---|
| HAYESAE | BIN | 2 | B | 4 |
| HAYESAE | DOC | 1 | A | 6 |
| MTERM4-7 | VIP | 1 | A | 3 |
| MTER8-11 | VIP | 1 | A | 3 |
| MTE12-17 | VIP | 1 | A | 5 |
| MTE18-21 | VIP | 1 | A | 3 |
| MTERMA1- | VIP | 1 | A | 4 |

**PD-10**

**COLOR COMP. FORTH**

| | | | | |
|---|---|---|---|---|
| MENU | BAS | 0 | B | 1 |
| FORTHMAN | UL1 | 2 | B | 7 |
| FORTHMAN | UL2 | 2 | B | 7 |
| FORTHMAN | UL3 | 2 | B | 7 |
| FORTH | BIN | 2 | B | 3 |
| OLDEDIT | DAT | 1 | A | 3 |
| FRTHDOC1 | TXT | 1 | A | 7 |
| FRTHDOC2 | TXT | 1 | A | 7 |
| FRTHDOC3 | TXT | 1 | A | 1 |
| EDIT | DAT | 1 | A | 3 |
| 32KFORTH | BIN | 2 | B | 4 |
| NEWFORTH | BIN | 2 | B | 3 |

**PD-11 MCPAINT**

**A COMPLETE GRAPHICS
DEVELOPMENT PROGRAM
WITH INSTRUCTIONS.**

| | | | | |
|---|---|---|---|---|
| RUN-ME | BAS | 0 | B | 1 |
| MCPAINT | BIN | 2 | B | 11 |
| ICONS | SYS | 2 | B | 3 |
| MCDOC | DOC | 1 | A | 11 |
| PRINTDOC | BAS | 1 | A | 1 |
| GLASDEMO | BIN | 2 | B | 6 |
| STARS | BIN | 2 | B | 2 |
| 1940S | SET | 2 | B | 1 |
| BLOON | SET | 2 | B | 1 |
| BOLD | SET | 2 | B | 1 |
| FANCY | SET | 2 | B | 1 |
| GREEK | SET | 2 | B | 1 |
| GREEKU | SET | 2 | B | 1 |
| HEBREW | SET | 2 | B | 1 |
| OLDENG | SET | 2 | B | 1 |
| TYPING | SET | 2 | B | 1 |
| EPSON | DRV | 2 | B | 1 |
| EPSON2 | DRV | 2 | B | 1 |
| ANIMATE | BAS | 0 | B | 1 |
| ANIMAT | BIN | 2 | B | 1 |
| BANNER | BAS | 0 | B | 2 |
| MCUTIL | BIN | 2 | B | 1 |

*** PD-12**

**PMODE 4 PICTURES**

**CHURCH, ROSES,
RUN"PIXFILES"JOY-
STICK IS REQUIRED**

| | | | | |
|---|---|---|---|---|
| XIXCMP | BAS | 0 | A | 3 |
| OUTPOST | BAS | 0 | A | 3 |
| OUTPOST | BIN | 2 | B | 3 |
| SFIELD | BAS | 0 | A | 2 |
| SFIELD | BIN | 2 | B | 3 |
| PIXFILES | BAS | 0 | B | 3 |
| TRUCK | BIN | 2 | B | 3 |
| MODEM | BIN | 2 | B | 3 |
| HORSE | BIN | 2 | B | 3 |
| MISSION | BIN | 2 | B | 3 |
| CLOISTER | BIN | 2 | B | 3 |
| RAIN | BIN | 2 | B | 3 |
| EAGLE | BIN | 2 | B | 3 |
| ROSES | BIN | 2 | B | 3 |
| CHURCH | BIN | 2 | B | 3 |
| GARDEN | BIN | 2 | B | 3 |
| PRES | BIN | 2 | B | 3 |
| LONI4 | BAS | 0 | A | 3 |

**PD-13**

**GRAPHICON PICTURE
DISK-1. REQUIRES
PIXFILES/BAS FROM
PD-12 & JOYSTICK**

| | | | | |
|---|---|---|---|---|
| PICTURES | GCM | 1 | B | 68 |

**PD-14**

**GRAPHICON PICTURE
DISK-2. REQUIRES
PIXFILES/BAS FROM
PD-12 & JOYSTICK**

| | | | | |
|---|---|---|---|---|
| PICTURES | GCM | 1 | B | 68 |

**PD-15**

**GRAPHICON PICTURE
DISK-3 REQUIRES
PIXFILES/BAS FROM
PD-12 & JOYSTICK**

| | | | | |
|---|---|---|---|---|
| PICTURES | GCM | 1 | B | 68 |

**PD-16**

**GRAPHICON PICTURE
DISK-4 REQUIRES
PIXFILES/BAS FROM
PD-12 & JOYSTICK**

| | | | | |
|---|---|---|---|---|
| PICTURES | GCM | 1 | B | 68 |

**PD-17 DISK UT.**

| | | | | |
|---|---|---|---|---|
| 64KBHW | BAS | 0 | A | 1 |
| AUTOSTRT | BAS | 0 | A | 1 |
| BAKDIR | BAS | 0 | A | 3 |
| BIN>BAS | BAS | 0 | A | 1 |
| CASSLABL | BAS | 0 | B | 1 |
| CURSOR | BAS | 0 | B | 1 |
| CUSTOM | BAS | 0 | B | 3 |
| CUSTOMIZ | BAS | 0 | B | 1 |
| DIR | BIN | 2 | B | 1 |
| DIR32 | BAS | 0 | A | 2 |
| DIR32C | DOC | 1 | A | 3 |
| DIRLISTR | BAK | 0 | B | 1 |
| DIRLISTR | BAS | 0 | B | 1 |

**PD-18  TAPE TO DISK
DISK UTILITIES**

| | | | | |
|---|---|---|---|---|
| DIRSORT | BAS | 0 | A | 1 |
| DISK-DIR | BAS | 0 | A | 1 |
| DISKLABL | BAS | 0 | A | 1 |
| LOADSOLU | BAS | 0 | B | 1 |
| MENU | BAS | 0 | B | 1 |
| PDIR | BAS | 0 | A | 1 |
| SORT | BAS | 0 | B | 1 |
| SORTPRT | BAS | 0 | B | 1 |
| SORTSAVE | BAS | 0 | A | 1 |
| SOULTION | BIN | 2 | B | 1 |
| SUPERBAC | BIN | 2 | B | 1 |
| T2D | BIN | 2 | B | 2 |
| TIMER | BAS | 0 | B | 1 |
| TPTODSK | BIN | 2 | B | 1 |

*** PD-19  GAMES**

| | | | | |
|---|---|---|---|---|
| 3DMAZE | BAS | 0 | A | 2 |
| BOXES | BAS | 0 | B | 1 |

| | | | | |
|---|---|---|---|---|
| CLOSE EN | BAS | 0 | B | 2 |
| CRITICAL | BAS | 0 | B | 1 |
| GAMMON | BAS | 0 | B | 3 |
| GOLDMINE | BAS | 0 | A | 3 |
| HOCKEY | BAS | 0 | A | 1 |
| HOGJOWL | BAS | 0 | A | 8 |
| HORSERAC | BAS | 0 | A | 3 |
| JUMPING | BAS | 0 | B | 1 |
| KALIDESC | BAS | 0 | B | 1 |
| MASTMIND | BAS | 0 | B | 1 |
| MEMORY | BAS | 0 | B | 1 |
| MOONBASE | BAS | 0 | B | 2 |
| NAMES | BAS | 0 | B | 4 |
| OTHELLO | BAS | 0 | B | 4 |

**PD-20  GAMES**

| | | | | |
|---|---|---|---|---|
| PEG | BAS | 0 | B | 3 |
| RABBIT | BAS | 0 | B | 1 |
| SAFE | BAS | 0 | B | 2 |
| SAUCER | BAS | 0 | B | 1 |
| SHOOTEM | BAS | 0 | B | 2 |
| SIMMON | BAS | 0 | A | 1 |
| SLITHER | BAS | 0 | A | 2 |
| SPACE WA | BAS | 0 | B | 4 |
| STAR TRE | BAS | 0 | B | 1 |
| SUBCHASE | BAS | 0 | B | 2 |
| SUBDESTR | BAS | 0 | B | 2 |
| SUNDANCE | BAS | 0 | A | 3 |
| TANKS | BAS | 0 | B | 2 |
| TOWER | BAS | 0 | B | 2 |
| UNDROVER | BAS | 0 | B | 1 |

**PD-21  MUSIC**

**PLAY MUSIC THROUGH
YOUR TV OR MONITOR.
COMPOSE,EDIT MUSIC.**

| | | | | |
|---|---|---|---|---|
| ORCH | BIN | 2 | B | 8 |
| ORCH | DOC | 1 | A | 3 |
| OCNVRT | BIN | 2 | B | 2 |
| GHOSBUST | MUS | 4 | M | 3 |
| STELMO | MUS | 4 | M | 2 |
| MASH | MUS | 4 | M | 2 |
| BOND1 | MUS | 4 | M | 2 |
| 2001 | MUS | 4 | M | 2 |
| ARIA | MUS | 4 | M | 2 |
| INVENTI | MUS | 4 | M | 1 |
| BATTSTAR | MUS | 4 | M | 2 |
| BOND2 | MUS | 4 | M | 2 |
| CLOSENCT | MUS | 4 | M | 2 |
| SCARBORO | MUS | 4 | M | 1 |
| FUGUEINC | MUS | 4 | M | 1 |
| MINUET | MUS | 4 | M | 1 |
| LONGTIME | MUS | 4 | M | 2 |
| MESSIAH | MUS | 4 | M | 3 |

*** PD-22  MUSIC-1**

**LOADM "NAME/MUS"
EXEC TO PLAY MUSIC
THROUGH TV OR MON.**

| | | | | |
|---|---|---|---|---|
| ADDPLAY | BAS | 0 | B | 1 |
| DEPLAY | BAS | 0 | B | 1 |
| MSQUEZ | BAS | 0 | B | 2 |
| ALSOSPAK | MUS | 2 | B | 5 |
| BOOGIE | MUS | 2 | B | 5 |
| CIRCUS | MUS | 2 | B | 2 |
| CLOWN | MUS | 2 | B | 2 |
| CLOWNS | MUS | 2 | B | 4 |
| HAYDEN | MUS | 2 | B | 8 |
| JBGOOD | MUS | 2 | B | 4 |
| PEACE | MUS | 2 | B | 5 |
| PEACH | MUS | 2 | B | 5 |
| PUFF | MUS | 2 | B | 6 |
| GOODDIEY | MUS | 2 | B | 4 |

## PD-23 MUSIC-2

```
LOADM "NAME/MUS"
EXEC TO PLAY MUSIC
THROUGH TV OR MON.

ADDPLAY   BAS 0 B 1
DEPLAY    BAS 0 B 1
MSQUEZ    BAS 0 B 2
RAIN      MUS 2 B 2
SONATA3   MUS 2 B 3
STRAV     MUS 2 B 4
FOGGY     MUS 2 B 4
FUNERAL   MUS 2 B 3
HARDDAY   MUS 2 B 2
INVENT    MUS 2 B 3
INVENT11  MUS 2 B 3
INVENT15  MUS 2 B 3
INVENT7   MUS 2 B 3
INVENT8   MUS 2 B 2
JOPLIN    MUS 2 B 4
KHAN      MUS 2 B 6
--------------------
```

## PD-24 MUSIC-3

```
LOADM "NAME/MUS"
EXEC TO PLAY MUSIC
THROUGH TV OR MON.

ADDPLAY   BAS 0 B 1
DEPLAY    BAS 0 B 1
MSQUEZ    BAS 0 B 2
PEANUTS   MUS 2 B 3
ROCK      MUS 2 B 5
ROXANNE   MUS 2 B 5
SCHERZO   MUS 2 B 5
TEACH     MUS 2 B 2
PIANOMAN  MUS 2 B 5
STRANGER  MUS 2 B 5
CAMELOT   MUS 2 B 4
CHACONNE  MUS 2 B 6
DIAMOND   MUS 2 B 3
DOWNROAD  MUS 2 B 4
FANTASY1  MUS 2 B 2
--------------------
```

## PD-25 MUSIC-4

```
LOADM "NAME/MUS"
EXEC TO PLAY MUSIC
THROUGH TV OR MON.

FANTASY2  MUS 2 B 3
GRENGRAS  MUS 2 B 4
HUMOR     MUS 2 B 4
INCROW    MUS 2 B 3
STARWARS  MUS 2 B 2
SUITEGM   MUS 2 B 6
SUPERMAN  MUS 2 B 2
WHENIM64  MUS 2 B 4
ROOTBEER  MUS 2 B 7
WAYUARE   MUS 2 B 3
AXELF     MUS 2 B 2
TOCATTA   MUS 2 B 3
--------------------
```

## PD-26 LAST WILL

```
LOAN      BAS 0 B 1
LASTWILL  BAS 0 B 6
IMEGA     BAS 0 B 3
AWARI     BAS 0 B 1
BACARAT   BAS 0 B 2
BAGELS    BAS 0 B 1
BLACKJAC  BAS 0 B 1
CHUCK     BAS 0 B 1
CONCENTR  BAS 0 B 1
CUBES     BAS 0 B 2
--------------------
```

## PD-27 GAMES

```
DEFUZE    BAS 0 B 1
DR ZEE    BAS 0 B 1
FLIPFLOP  BAS 0 B 1
GO-FISH   BAS 0 B 2
HANGMAN   BAS 0 B 2
```

---

```
HIGHLOW   BAS 0 B 1
JACKPOT   BAS 0 B 1
KEYS      BAS 0 B 1
L & M     BAS 0 B 3
LUNARLD   BAS 0 B 2
NUMBERS   BAS 0 B 1
OBSTACLE  BAS 0 B 1
POOLGAME  BAS 0 B 4
RETURN    BAS 0 B 1
REVERSI   BAS 0 B 2
STARTREK  BAS 0 B 2
TTREK     BAS 0 B 3
--------------------
```

## PD-28 COMM. CC-TALK, BBS, TERM

```
BBS'S     DAT 1 A 1
CCT       IO  2 B 1
CCTALK    BAS 0 B 1
CNFG40V1  BAS 0 A 5
CNFG40V2  BAS 0 A 4
CTLKEY    BAS 1 A 1
MTERM1    DOC 1 A 11
MTERM2    DOC 1 A 8
MTERM40   BIN 2 B 8
REDIAL    BAS 0 A 1
PACREDIA  BAS 0 A 1
--------------------
```

## PD-29 COMM, WORD PRO, GAMES

```
GOSTSHIP  BAS 0 B 8
INT RATE  BAS 0 B 2
INVSTANL  PC  0 B 4
MENU      BAS 0 B 4
MOTOJUMP  BAS 0 B 3
SCREEN    MAX 2 B 6
SCREEN1   BIN 2 B 3
SCREEN2   BIN 2 B 3
SCREEN2   MAX 2 B 6
STRINGTU  BAS 0 B 4
TTERM     DSK 2 B 4
TTHELP    DAT 1 A 4
USING     BAS 0 B 3
WF-DOC    JP  0 B 2
WORDFILE  JP  0 B 4
PARM1     DAT 1 A 1
--------------------
```

## PD-30 CHECK BOOK, UTILITIES

```
CHECKBOK  BAS 0 B 4
CHECKBOK  DOC 1 A 9
DIRR      CMD 2 B 1
DVIEW     BAS 0 B 1
FILEMAID  BAS 0 B 2
LISTER    BAS 0 B 1
PAINTPOT  BAS 0 B 4
SCREEN    MAX 2 B 6
SCREEN1   BIN 2 B 3
SCREEN2   BIN 2 B 3
SCREEN2   MAX 2 B 6
SPECZAP   BAS 0 B 5
TAPETYPE  BIN 2 B 1
TTERM     DSK 2 B 4
DVIEW     DSK 0 B 1
MENU      BAS 0 B 4
--------------------
```

## PD-31

PIRATES TREASURE - As you explore the cave looking for the treasure, a picture appears on the screen as you go from room to room. These pictures are loaded from disk. A computer with a disk drive is required and a ramdisk is preferred.

--------------------

## PD-32

Color Computer 3 moving pictures. Consists of a beautiful waterfall and a colorful bouncing ball.

```
WATRFALL  BAS 0 B 1
WATRFALL  BIN 2 B 1
WATRFALL  MGE 1 B
BALL      BAS 0 B 1
BALL2     BAS 0 B 1
BOUNCE    BIN 2 B 1
BALL2     HR1 2 B 4
BALL2     HR2 2 B 4
BALL2     HR3 2 B 4
BALL2     HR4 2 B 4
--------------------
```

## PD-33

EDUCTIONAL PROGRAMS

```
ABBREV    BAS 0 B 4
ABCPOP    BAS 0 B 3
ALPHAAL   BAS 0 B 1
EDUCATE   BAS 0 B 1
HANGP     BAS 0 B 1
HOMONYM   BAS 0 B 1
SPELWORD  BAS 0 B 1
MATH      BAS 0 B 2
DRILL     BAS 0 B 2
MLTP      BAS 0 B 1
ROUND     BAS 0 B 2
AREA      BAS 0 B 5
METCONV   BAS 0 B 3
NUMBERS   BAS 0 B 2
SIEVE     BAS 0 B 1
--------------------
```

## PD 36

COMP.SCIENCE PGMS 1
These programs are tutorials on basic programming.

```
COMPSC1   BAS 0 B 8
COMPSC2   BAS 0 B 3
COMPSC3   BAS 0 B 9
COMPSC4   BAS 0 B 5
COMPSC5   BAS 0 B 9
COMPSC6   BAS 0 B 5
GETPUT    BAS 0 B 2
--------------------
```

## PD 37

COMP.SCIENCE PGMS 2

These programs are tutorials on basic programming.

```
IFTHEN    BAS 0 B 9
EXTENDED  BAS 0 B 2
GETPUT    BAS 0 B 2
COMPSCI8  BAS 0 B 8
COMPSCI9  BAS 0 B 5
COMPSCI7  BAS 0 B 7
EXTDEMO   BAS 0 B 3
--------------------
```

## PD 38

EDUCATIONAL PGMS
These programs are excellent learning tools for school children.

```
ABBREV    BAS 0 B 4
ABCPOP    BAS 0 B 3
ALPHAAL   BAS 0 B 1
```

---

```
EDUCATE   BAS 0 B 1
HANGP     BAS 0 B 1
HOMONYM   BAS 0 B 1
SPELWORD  BAS 0 B 2
MATH      BAS 0 B 2
DRILL     BAS 0 B 2
MLTP      BAS 0 B 1
ROUND     BAS 0 B 2
AREA      BAS 0 B 5
METCONV   BAS 0 B 3
NUMBERS   BAS 0 B 2
```

## PD 39

ADDRESS FILES AND FINANCE PROGRAMS

```
PHONE     BAS 0 B 1
LABELPRT  BAS 0 B 1
LETTER    BAS 0 B 3
MAILIST   BAS 0 B 1
WORDPROC  BAS 0 B 3
MAILLST   BAS 0 B 2
PHONLST   BAS 0 B 1
MINIWORD  BAS 0 B 2
LNWIDTH   BAS 0 B 1
CHKWRITE  BAS 0 B 2
CHKANAL   BAS 0 B 4
PRNTCHK   BAS 0 A 1
CHECKS    BAS 0 B 4
CHCKSTUB  BAS 0 B 1
TOTALS    DAT 1 A 1
CHECKS    DAT 1 A 1
GRAPH     BAS 0 B 4
LOAN      BAS 0 B 3
CALC      BAS 0 B 1
PAYMENT   BAS 0 B 1
CASHJNL   BAS 0 B 3
AMORT     BAS 0 B 3
```

## PD-40

TAPE-DSK & DSK-TAPE
With these programs you can copy a disk to tape or a tape to disk.

```
T2D       BIN 2 B 2
DTCOPY    BIN 2 B 1
DSK-TP    BAS 0 B 1
DISKLIST  BAS 0 B 3
DIRLIST   BAS 0 B 2
DISKDUMP  BAS 0 B 1
CASSDIR   BAS 0 B 1
```

## PD-41
Picture files

```
STAMPS    MAX 2 B 3
STARTREK  MAX 2 B 3
ST-TREK2  MAX 2 B 3
SCHOOL    MAX 2 B 3
SATURN    MAX 2 B 3
ESCHER    MAX 2 B 3
LABOR     MAX 2 B 3
MASK      MAX 2 B 3
BUG BOX   MAX 2 B 3
SPACE     MAX 2 B 3
EASTER    MAX 2 B 3
SPACE 2   MAX 2 B 3
POPEYE    MAX 2 B 3
GARFIELS  MAX 2 B 3
BEETLE B  MAX 2 B 3
POLO      MAX 2 B 3
MAGAR     MAX 2 B 3
X-PAD     MAX 2 B 3
CASTLE    MAX 2 B 3
MUSIC TV  MAX 2 B 3
COCO      MAX 2 B 3
--------------------
```

## PD-42
Picture files

```
TITLES    MAX 2 B 3
PIXFILES  BAS 0 B 3
THOLIAN   MAX 2 B 3
```

---

```
3001AD    MAX 2 B 3
F15       MAX 2 B 3
QUEEN     MAX 2 B 3
BRONCOS   MAX 2 B 3
STARTREK  MAX 2 B 3
ROOM      MAX 2 B 3
RAMBO     MAX 2 B 3
OWL       MAX 2 B 3
ENTERPR   MAX 2 B 3
STAR-T3   MAX 2 B 3
NCC-1701  MAX 2 B 3
SAT-2     MAX 2 B 3
ATMOSP    MAX 2 B 3
STARWARS  MAX 2 B 3
ORIENTAL  MAX 2 B 3
--------------------
```

## PD-43
Picture files

```
STAMP     MAX 2 B 3
STRIPE    MAX 2 B 3
WOMAN     MAX 2 B 3
BLUEJAY   MAX 2 B 3
LUCY      MAX 2 B 3
OLD ENG   MAX 2 B 3
MENU1     MAX 2 B 3
OWL       MAX 2 B 3
VAN GOG   MAX 2 B 3
WOMAN1    MAX 2 B 3
PSH       MAX 2 B 3
DUCKPOND  MAX 2 B 3
RANGER    MAX 2 B 3
PLANET    MAX 2 B 3
CHRSTMAS  MAX 2 B 3
PEACE     MAX 2 B 3
WOMAN3    MAX 2 B 3
HAWK      MAX 2 B 3
PHASER    MAX 2 B 3
PIXFILES  BAS 0 B 3
--------------------
```

## PD-44

Terminal pgm with documentation. This will work with the CoCo-3. Instructions are included.

```
MTRM43    BIN 2 B 8
CONFIG43  BAS 0 B 4
MTSTART   BAS 0 B 4
MTERM1    DOC 1 A 11
MTERM2    DOC 1 A 8
MTERM3    DOC 1 A 7
DOS BOOT  DAT 1 A 1
*         *   0 B 1
**        *   1 A 1
READDOC   BAS 0 B 1
--------------------
```

## PD-45
Picture Files

```
DRAGON    MAX 2 B 3
HOT LIPS  MAX 2 B 3
ANIMALS   MAX 2 B 3
CLOWN F   MAX 2 B 3
FISH      MAX 2 B 3
3 MEN     MAX 2 B 3
S MAP     MAX 2 B 3
BUGS      MAX 2 B 3
CFISH     MAX 2 B 3
HERO      MAX 2 B 3
WMAP      MAX 2 B 3
GSCOTT    MAX 2 B 3
STATES    MAX 2 B 3
HORSE     MAX 2 B 3
CROSS     MAX 2 B 3
FOODW     MAX 2 B 3
RSTONE    MAX 2 B 3
COCO      MAX 2 B 3
ALIEN     MAX 2 B 3
PIXFILES  BAS 0 B 3
--------------------
```

**Column 1**

```
* PD-46
Talk and Music
Files (C)LOADM
"FILE" then EXEC.

TALK      BIN 2 B 11
TALK2     BIN 2 B 11
WILLTELL  BIN 2 B 9
MUSICBOX  BIN 2 B 1
BEATLES   BIN 2 B 4
JUMP      BIN 2 B 5
GRELN     BIN 2 B 5
GHOST     BIN 2 B 5
JINGLE    BIN 2 B 3
WORLD     BIN 2 B 5
CTRYROAD  BIN 2 B 2
--------------------
* PD-47

Miscellaneous Pgms

T         BAS 0 B 2
SANTEE2   BAS 0 B 1
MILEAGE   BAS 0 B 1
M         BAS 0 B 1
DIGITS    BAS 0 B 1
NUMBLIST  BAS 0 B 1
COUNT     BAS 0 B 1
SC        BAS 0 B 1
DRAWTEXT  BAS 0 B 1
SAMPLE    BAS 0 B 1
GRSCRWRT  BAS 0 B 2
HRTEXT2   BAS 0 B 3
DRAW      BAS 0 B 1
WRITER    BAS 0 B 1
TYPEBET   BAS 0 B 2
WRITEBET  BAS 0 B 2
TEXT2     BAS 0 B 2
SANTEE    BAS 0 B 2
SHUTTLE   BAS 0 B 1
AJOCK     BAS 0 B 1
PLATFORM  BAS 0 B 1
MAZE      BAS 0 B 4
DISKZAPR  BAS 0 B 2
ZAP       BAS 0 B 3
DETHSHIP  BAS 0 B 3
BACKUP35  BAS 0 B 1
BOOT      BAS 0 B 1
SCRNLIST  BAS 0 B 1
DOSSTART  BAS 0 B 1
LABEL     BAS 0 B 2
DSKDSABL  BAS 0 B 1
NOFREEG   BAS 0 B 1
FORMATER  BAS 0 B 1
ROMRAM    BIN 2 B 1
SUPDUP    BIN 2 B 1
TESTTEXT  BAS 0 B 1
--------------------
* PD-48

Miscellaneous Pgms

EXTBAS    BAS 0 B 3
DISAPEAR  BAS 0 B 1
PAINT     BAS 0 B 1
DATA      BIN 2 B 1
DATA2     BIN 2 B 1
SCRDATA   BIN 2 B 1
FILL2     BIN 2 B 2
QUADDRAW  BAS 0 B 1
CELTIC    BAS 0 B 2
ALL RAM   BAS 0 B 1
CHARGEN   BIN 2 B 1
ROMRAM    BIN 2 B 1
OBSTACLE  BAS 0 B 1
64K RAM   BAS 0 B 1
COLORSEL  BAS 0 B 1
TRIG      BAS 0 B 4
ALGEBRA   BAS 0 B 4
PLAY      BAS 0 B 1
STATECAP  BAS 0 B 2
MLSOUNDS  BAS 0 B 1
```

**Column 2**

```
ROTATION  BAS 0 B 2
PARABOLA  BAS 0 B 2
INSTAPIC  BAS 0 B 1
CLOVER    BAS 0 B 1
HAT-PLOT  BAS 0 B 1
WHEEL 1   BAS 0 B 1
LETTER-R  PAR 1 A 1
3-LINES   ROT 1 A 1
TRAPZOID  ROT 1 A 2
PYRAMID   ROT 1 A 2
CUBE      ROT 1 A 3
51X24     BAS 0 B 2
WINDOW    BAS 0 B 5
GGPRTSU   BAS 0 B 1
KALEIDO   BAS 0 B 1
OK83APRT  BAS 0 B 1
NUMCNVTR  BAS 0 B 1
ADVRTN    BAS 0 B 1
--------------------
* PD-49

Miscellaneous Pgms.

BC        BIN 2 B 10
PEDRO     BIN 2 B 11
BLOCKADE  BAS 0 B 3
REPEAT    BAS 0 B 1
AIRPLANE  BAS 0 B 1
BUSTOUT   BAS 0 B 1
GOLF      BAS 0 B 7
CITY      BAS 0 B 2
AIR-RAID  BAS 0 B 2
MAZE      BAS 0 B 4
DUALDUP   BIN 2 B 2
DIRMAP    BAS 0 B 3
CHESS     BAS 0 B 5
WHATZIT   BAS 0 B 4
BATLSHIP  BAS 0 B 3
SP*ROCKS  BAS 0 B 1
--------------------
* PD-50

Miscellaneous PGMS

GOBBLER   BAS 0 B 2
PYTHON    BAS 0 B 2
LUNAR     BAS 0 B 2
LUNALANA  BAS 0 B 1
AMAZING   BAS 0 B 2
BALLOON   BAS 0 B 1
VAPORWRM  BAS 0 B 2
ABM       BAS 0 B 3
BULLSEYE  BAS 0 B 1
CRASH     BAS 0 B 1
DOTS      BAS 0 B 3
F-16      BAS 0 B 3
KRYPTON   ART 2 B 3
KRYPTON   BAS 0 B 1
KRYPTON   GAM 0 B 1
NUKEATTK  BAS 0 B 2
ASTEROID  BAS 0 B 1
PRIX      BAS 0 B 3
ONE       BIN 2 B 3
TWO       BIN 2 B 3
THREE     BIN 2 B 3
FOUR      BIN 2 B 3
TEMPEST   BAS 0 B 2
SNAKE     BAS 0 B 2
SCORE     DAT 1 A 1
OTHELLO   BAS 0 B 4
ROCKS     BAS 0 B 3
LANDER    BAS 0 B 2
--------------------
* PD-51
Games & Programs

DRAGRACE  BAS 0 B 1
WORMER    BAS 0 B 2
SIMON     BAS 0 B 2
RIDER     BAS 0 B 2
MISSILE   BAS 0 B 3
```

**Column 3**

```
LETSHOOT  BAS 0 B 2
SHOOTGAL  BAS 0 B 2
MISSILE2  BAS 0 B 3
FENCE     BAS 0 B 3
BANDIT    BAS 0 B 1
CHICKEN   BAS 0 B 2
MAXIMUM   BAS 0 B 3
FLIGHT    BAS 0 B 2
COVERUP   BAS 0 B 2
WORLDMAP  BAS 0 B 4
POUNCE    BAS 0 B 1
MARTIANS  BAS 0 B 2
FINDIT    BAS 0 B 3
SCRAMBLE  BAS 0 B 5
BOUNBABY  BAS 0 B 2
CHICK     BAS 0 B 3
BOBO      BAS 0 B 3
RUBIC     BAS 0 B 4
MCJUMP    BAS 0 B 3
--------------------
* PD-52
Picture files

COCO      MAX 2 B 6
COL COCO  MAX 2 B 6
MOOSHEAD  MAX 2 B 6
COKE      MAX 2 B 6
CUBS      MAX 2 B 6
REDS      MAX 2 B 6
BREAKERS  MAX 2 B 6
USFL      MAX 2 B 6
SPACE     BIN 2 B 3
GIZMO     MAX 2 B 3
DINASOUR  MAX 2 B 3
--------------------
* PD 53
Picture Files

INDIAN    MAX 2 B 6
HOMECOME  MAX 2 B 6
GRIN      BIN 2 B 3
TARD      BIN 2 B 3
STUD      BIN 2 B 3
COMET     BIN 2 B 3
DESERT    BIN 2 B 3
FOOD      BIN 2 B 3
SMIRK     BIN 2 B 3
PLAYA     BIN 2 B 3
HELLO     BIN 2 B 3
GROVER    BIN 2 B 3
DRIVE IN  BIN 2 B 3
TIME      BIN 2 B 3
KOALA     BIN 2 B 3
PATTERN   BIN 2 B 3
HAGAR     BIN 2 B 3
CHIPS     BIN 2 B 3
--------------------
* PD 54
Picture Files

PENTAGON  PIC 2 B 3
GRID 2    PIC 2 B 3
SNOWFLAK  PIC 2 B 3
CONETUNL  PIC 2 B 3
4-POINT   PIC 2 B 3
BALTSTR   MAX 2 B 3
CARTOON   MAX 2 B 3
HUELEWIS  MAX 2 B 3
STARTREK  MAX 2 B 3
HOUSE1    MAX 2 B 6
HOUSE2    MAX 2 B 6
LIFECYCL  MAX 2 B 6
COCOMAG   MAX 2 B 3
MASCASTL  MAX 2 B 3
COLUMBIA  MAX 2 B 3
POLO      MAX 2 B 3
ET        BAS 0 B 7
WHEEL 1   PIC 2 B 3
--------------------
```

**Column 4**

```
* PD-55
Picture Files

PARKERPT  MAX 2 B 3
TOWER     PIC 2 B 3
TOWER2    PIC 2 B 3
SCREEN    PIC 2 B 3
BOMB      PIC 2 B 3
ANDRON    PIC 2 B 3
SALE      PIC 2 B 3
CHIPS     PIC 2 B 3
TUNLROAD  BIN 2 B 3
LONEROAD  BIN 2 B 3
CITYROAD  BIN 2 B 3
LAKEROAD  BIN 2 B 3
CROSROAD  BIN 2 B 3
BLACK     BIN 2 B 3
CAL1      BIN 2 B 3
CAL2      BIN 2 B 3
CAL3      BIN 2 B 3
3-LEAF    PIC 2 B 3
5-STARS   PIC 2 B 3
SPHERE    PIC 2 B 3
15-LEAF   PIC 2 B 3
--------------------
* PD-56

Glossary, Memory
Maps, Programs

COCO      VIP 1 A 4
VIP ON 3  VIP 1 A 1
BEEF      VIP 1 A 1
MCTRM3    VIP 1 A 1
GLOSSARY  VIP 1 A 7
POKEPEEK  VIP 1 A 17
WIDTH     VIP 1 A 1
COCO 3    VIP 1 A 17
MISSLES   BAS 0 B 2
CLOCK     BAS 0 B 1
JET       BAS 0 B 4
--------------------
* PD-57
Picture Files

VAMPIRE   PIC 2 B 3
ATLANTA   BAS 0 B 3
NOGHOST   PIC 2 B 3
AIRPORT   BAS 0 B 4
S EASTON  BAS 0 B 4
1SMLSTEP  BAS 0 B 4
HAGAR     PIC 2 B 3
SUNSET    BAS 0 B 3
S NICKS   BAS 0 B 4
SNOOPY1   BAS 0 B 3
MICKEY    BIN 1 B 8
DONALD    BIN 2 B 8
SNOOPY2   BAS 0 B 4
SNOOPY3   BAS 0 B 4
SNOOPY4   BAS 0 B 4
--------------------
* PD-58
Miscellaneous Pgms.

DISKLIST  BAS 0 B 1
DIRLIST   BAS 0 B 2
ML ADDR   BAS 0 B 1
DISKDUMP  BAS 0 B 1
PRINUTIL  BAS 0 B 2
CALPRINT  BAS 0 B 3
ALPHSONG  BAS 0 B 1
PAINT     BAS 0 B 1
DOGPICT   BAS 0 B 2
EVADER    BAS 0 B 1
NUKATTC   BAS 0 B 2
BASICMAP  BAS 0 B 3
JOYPAINT  BAS 0 B 1
PUMPKIN   BAS 0 B 1
HOMOYMS   BAS 0 B 1
ABBREV    BAS 0 B 4
```

**Column 5**

```
CONVERT   BAS 0 B 3
CASSDIR   BAS 0 B 1
CVERT     BAS 0 B 1
FLASCARD  BAS 0 B 1
MESSAGE   BAS 0 B 1
RELOCAT   BAS 0 B 1
COUNT     BAS 0 B 1
CALENDAR  BAS 0 B 1
DOGS      BAS 0 B 1
DOGFIGHR  BAS 0 B 1
BEAST     BAS 0 B 1
--------------------
* PD-59
GAMES, UTILITIES

64X64F    BAS 0 B 1
RND#'S    BAS 0 B 1
SCROLLER  BAS 0 B 1
COCOBUG   BAS 0 B 2
DRWBOARD  BAS 0 B 1
SPACE     BAS 0 B 1
DIR-ADDR  BAS 0 B 1
BACKGAMN  BIN 2 B 2
CHESS     BIN 2 B 3
BATTLE    BIN 2 B 2
GERM      BIN 2 B 1
BLEEP     BAS 0 B 3
TICKER    BAS 0 B 3
LEAKYTAP  BAS 0 B 3
UTOPIAN   BAS 0 B 4
COLORDOT  BAS 0 B 3
STAYALIV  BAS 0 B 3
TIMEFLT   BAS 0 B 3
NAVGUNS   BAS 0 B 2
ATACMAN   BAS 0 B 3
CALENDAR  BAS 0 B 1
POKER25   BAS 0 B 2
VIEWER5   BAS 0 B 1
STUFF     BAS 0 B 1
--------------------
* PD 60 Basic Pgms

S NICKS   BAS 0 B 4
1SMLSTEP  BAS 0 B 4
SUNSET    BAS 0 B 3
3DTTT     BAS 0 B 4
BATTSHIP  BAS 0 B 2
CRACE     BAS 0 B 2
FLY       BAS 0 B 3
KINGS     BAS 0 B 6
KINGTUT   BAS 0 B 7
OREGON    BAS 0 B 9
POKER     BAS 0 B 2
ROBOTS    BAS 0 B 3
ROLLON    BAS 0 B 2
SORCERER  BAS 0 B 6
--------------------
* PD-61 Pictures

HAGAR     PIC 2 B 3
SHIPS     BAS 0 B 2
SIGNS5    BAS 0 B 1
SPACE     BAS 0 B 8
3GUYS     MAX 2 B 3
AIRPORT   BIN 2 B 6
BIGCAT    MAX 2 B 3
CUBE      BIN 2 B 3
DOGPICT   BAS 0 B 2
EARTH     MAX 2 B 3
GARFIELD  PIX 2 B 3
GIRL      MAX 2 B 3
NEWWAVE   MAX 2 B 3
OLIVER    MAX 2 B 3
OWL       MAX 2 B 3
PEANUTS   PIX 2 B 3
SHUTTLE   MAX 2 B 3
SR-71     MAX 2 B 3
ZEBCHESS  MAX 2 B 3
ZIGGY     PIX 2 B 3
--------------------
```

```
* PD- 62 Basic Pgms      HILLST   BIN 2 B 4      PD-69 Disk Utilities    Machine Lang. Pgms      ASSEMBLR BAS 0 B 3
                         -------------------                             FIND     BAS 0 B 1      ALLKEYS  BAS 0 B 2
ALARM    BAS 0 B 2                               DIRPBR   BAS 0 B 2      LOCFIND  BAS 0 B 1      ALPHABET BAS 0 B 1
BIBLE    BAS 0 B 2       * PD-66 Basic Pgms      DISKLOOK BAS 0 B 1      ML ADDR  BAS 0 B 1      BLKBOARD BAS 0 B 4
BINGOCD  BAS 0 B 1                               DKTODK   BAS 0 B 3      MLFINDER BAS 0 B 1      CHAR-GEN BAS 0 B 1
CHECKS   BAS 0 B 3       64KMEMT  BAS 0 B 2      DSK2TP   BAS 0 B 2      MLTTD    BAS 0 B 1      CREATOR  BAS 0 B 1
CLOCK    BAS 0 B 1       AUTODIAL BAS 0 B 2      DSKLIBRY BAS 0 B 3      READBIN  BAS 0 B 1      EXEC LET BAS 0 B 1
DATA3    BAS 0 B 3       FINDAWRD BAS 0 B 2      DSKMSTER BAS 0 B 4      RELOCAT  BAS 0 B 1      GRLETTER BAS 0 B 1
DATES    BAS 0 B 2       FLASHCRD BAS 0 B 2      DSKSPEED BAS 0 B 1      CHKBOOK  BAS 0 B 3      LOWRCASE BAS 0 B 2
DECIDE   BAS 0 B 3       PHONEWRD BAS 0 B 1      DTOD     BAS 0 B 2      FINANAD  BAS 0 B 6      NUMBERS  BAS 0 B 2
EXREF    BAS 0 B 3       64KTEST  BAS 0 B 1      D UTIL   BAS 0 B 2      GRAPHICS BAS 0 B 5      ROTO14   BAS 0 B 2
FILES    BAS 0 B 4       ABBREV   BAS 0 B 4      DDCOPY   BAS 0 B 2      HOMEUTIL BAS 0 B 6      ROTATE   BIN 2 B 4
FLIPPAGE BAS 0 B 3       BASECONV BAS 0 B 1      DIRGET   BAS 0 B 2      LIFE     BAS 0 B 4      SMALLBET BAS 0 B 1
LABELPRT BAS 0 B 1       BIORYTHM BAS 0 B 3      DIRLIST  BAS 0 B 2      MCONVERT BAS 0 B 2      DOT      BAS 0 B 1
MESSAGE  BAS 0 B 1       BOWLSUM  BAS 0 B 2      DIRSAVE  BAS 0 B 1      METCONV  BAS 0 B 1      HI!      BAS 0 B 1
OFFSET   BAS 0 B 1       BOXLABEL BAS 0 B 1      DISK FIX BAS 0 B 1      JOYLIST  BAS 0 B 1      EYES     BAS 0 B 1
PHONE    BAS 0 B 1       CALENDAR BAS 0 B 2      DISKDIRE BAS 0 B 3      CLOCK    BIN 2 B 1      YOYO     BAS 0 B 1
PHONEDIR BAS 0 B 2       CALENDR2 BAS 0 B 2      DISKDUMP BAS 0 B 1      CAMELOT  BIN 2 B 2      SCROLE   BAS 0 B 1
PILOT    BAS 0 B 2       CAR CALC BAS 0 B 1      DISKEDIT BAS 0 B 4      FIRE     BIN 2 B 6      SCREEN   PIC 2 B 3
PROJEVAL BAS 0 B 4       CASSDIR  BAS 0 B 1      DISKLIST BAS 0 B 1      CLOCK    DAT 1 A 1      -------------------
SPELWORD BAS 0 B 1       CONTOUR  BAS 0 B 1      DISKSORT BAS 0 B 1      -------------------
VALENCE  BAS 0 B 2       CONVERGE BAS 0 B 1      DISKTEST BAS 0 B 1                             PD-76 Misc. Pgms.
-------------------      CONVERT  BAS 0 B 3      DISKTIME BAS 0 B 1      * PD 73 Basic Pgms
                         COUNT    BAS 0 B 1      DSKCLEAN BAS 0 B 1                             SPCHK    BAS 1 A 1
* PD-63 Basic Pgms       CVERT    BAS 0 B 1      MASTRDSK BAS 0 B 4      CARTEL   BAS 0 B 7      SPELCONT BAS 1 A 2
                         DEC<>HEX BAS 0 B 1      -------------------     DODGE-EM BAS 0 B 2      ALFABETA BAS 0 B 2
ANIMALS  BAS 0 B 3       FUELCOST BAS 0 B 1                              DOGS     BAS 0 B 1      ALPHABET BAS 0 B 3
BALOONS  BAS 0 B 3       HEXLOAD  BAS 0 B 1      * PD-70  Basic Pgms     DOORS    BAS 0 B 1      AUTOLINE BAS 0 B 1
BATSHIP  BAS 0 B 4       HEXTODEC BAS 0 B 1                              PINGPONG BAS 0 B 1      CCMETEOR BAS 0 B 3
BUGS     BAS 0 B 6       IN-OUT   BAS 0 B 1      MLADFND  BAS 0 B 2      CACAPHON BAS 0 B 1      CENTRIT  BAS 0 B 3
CONNECT4 BAS 0 B 4       HOMONYMS BAS 0 B 1      BIGHILL  BAS 0 B 1      SUB      BAS 0 B 5      COCOMIND BAS 0 B 2
DIGGEM   BAS 0 B 3       JOYPAINT BAS 0 B 1      BLACKJK  BAS 0 B 4      SURVIVAL BAS 0 B 5      ERAZOR   BIN 2 B 4
FACTORS  BAS 0 B 4       KALVOS   BAS 0 B 1      CIA      BAS 0 B 6      TREK     BAS 0 B 5      FLAGQUIZ BAS 0 B 4
GEOGAME  BAS 0 B 4       LINES    BAS 0 B 1      CIPHER   BAS 0 B 1      TYCOON   BAS 0 B 2      GARDEN   BAS 0 B 4
KINGDOM  BAS 0 B 6       MACDATA  BAS 0 B 1      CUBES    BAS 0 B 1      SCRAMBLE BAS 0 B 5      KT MOVES BAS 0 B 2
MAZE3    BAS 0 B 3       MISSLETT BAS 0 B 1      DOGFIGHT BAS 0 B 1      SIMON    BAS 0 B 1      LINECTRL BAS 0 B 2
MISSILES BAS 0 B 2       -------------------     FISH     BAS 0 B 1      WHERISIT BAS 0 B 2      POKRMACH BAS 0 B 2
POKER    BAS 0 B 4                               FLIP     BAS 0 B 2      WALLHIT  BAS 0 B 1      PRINTDAT BAS 0 B 2
SLOTS    BAS 0 B 2       * PD-67 Basic Pgms      FOOTBALL BAS 0 B 4      TICTACT  BAS 0 B 2      PUNTER   BAS 0 B 4
TROLL    BAS 0 B 6                               GOLDMINE BAS 0 B 3      CHBASIC  BAS 0 B 1      RALLY    BIN 2 B 4
-------------------      LOAN     BAS 0 B 3      HANGMAN  BAS 0 B 2      -------------------     RBASIC   BIN 2 B 1
                         LOANAMOR BAS 0 B 1      HILOW    BAS 0 B 3                              RS       TXT 1 A 1
* PD-64 Basic Pgms       64KLOOK  BAS 0 B 8      HOBBIT   BAS 0 B 2      PD-74 Basic Pgms       RS       BIN 2 B 1
                         ASSEMBLR BAS 0 B 3      HUSTLE   BAS 0 B 1                              RS       BAS 0 A 1
OHMSLAW   BAS 0 B 1      DISASSY  BAS 0 B 4      JUMP     BAS 0 B 1      SATAN    BAS 0 B 8      SCUBA    BAS 0 B 3
POWER UP  BAS 0 B 1      FINANCE  BAS 0 B 8      MEMORY   BAS 0 B 2      CHESS    BAS 0 B 5      SPELL    P1  1 A 1
ROMPACK   BAS 0 B 1      ROMDUMP  BAS 0 B 1      PROTECT  BAS 0 B 2      CIPHER   BAS 0 B 1      USE64K   BIN 2 B 1
ROMRAM    BAS 0 B 1      WEREWAND BAS 0 B 5      QUEST    BAS 0 B 4      CONTROL  BAS 0 B 5      USE64KB  BAS 0 B 1
SCRDUMP   BAS 0 B 1      CHECKS   BAS 0 B 4      SLITHER  BAS 0 B 1      DECTOHEX BAS 0 B 1      USE64KS  BAS 0 B 2
SLOSKROL  BAS 0 B 1      MONEYHLP BAS 0 B 4      STOCK    BAS 0 B 3      TRUCKER  BAS 0 B 6      ZAP      BAS 0 B 3
SORT      BAS 0 B 1      CHKBOOK  BAS 0 B 3      -------------------     DSKMAIL  BAS 0 B 2      SPELLER  DOC 1 A 2
SPEDMATH  BAS 0 B 3      STAT-LOG BAS 0 B 3                              FREEWAY  BAS 0 B 1      -------------------
SPOOLER   BIN 2 B 1      WORDPRC  BAS 0 B 5      * PD-71 Basic &         FUELCOST BAS 0 B 1
UPPER32K  BAS 0 B 1      WORDSCAR BAS 0 B 2        Machine Lang. Pgms    GERMS    BAS 0 B 2      PD-77 Basic Pgms
STRIKE    BAS 0 B 1      TYPING   BAS 0 B 2                              ITALIC   BAS 0 B 2
SHIPS     BAS 0 B 2      -------------------     DISASSEM BAS 0 B 2      OLDHOUSE BAS 0 B 5      BALLOON  BAS 0 B 3
WILLSADV  BAS 0 B 5                              PAYMENT  BAS 0 B 1      PRINTEST BAS 0 B 1      BATLSHIP BAS 0 B 6
RACEWAY   BAS 0 B 4      * PD-68 Basic Pgms      STATCAP  BAS 0 B 2      REFLEX   BAS 0 B 1      COCOART  BAS 0 B 3
TREK      BAS 0 B 4                              TEMPCONV BAS 0 B 1      SECTDUMP BAS 0 B 1      ESP TEST BAS 0 B 3
TXTCNVRT  BAS 0 B 1      ART      BAS 0 B 1      ECHOSONG BAS 0 B 1      SILLY    BAS 0 B 2      GOSFILL  BAS 0 B 1
-------------------      BARGRAPH BAS 0 B 1      MUSCONV  BAS 0 B 1      STATECAP BAS 0 B 2      GOSPER   BAS 0 B 1
                         BEGIN    BAS 0 B 1      FUGUE    BIN 2 B 3      SONGDISK BAS 0 B 2      HILBERT  BAS 0 B 3
* PD-65 Music            BWDUMP   BIN 2 B 1      JETI     BIN 2 B 3      STEVEHAR BAS 0 B 1      HOMEBUDG BAS 0 B 3
                         CHAR     BAS 0 B 2      MOONLIT  BAS 2 B 3      SUB      BAS 0 B 2      KANGA    BAS 0 B 1
MUSIC     BIN 2 B 7      COM      BAS 0 B 2      RONDO    BIN 2 B 4      TANKS    BAS 0 B 2      LEDGER   BAS 0 B 3
MUSIC1    BAS 0 B 1      DISMON   BAS 0 B 7      LOOKLOVE BIN 2 B 1      WATOR    BAS 0 B 3      LINE ART BAS 0 B 1
SOUND     ASM 1 A 1      DOT      BAS 0 B 1      MENU     BAS 0 B 1      WESTERN  BAS 0 B 3      LOAN     BAS 0 B 3
SOUNDDEM  BAS 0 B 1      EDITOR   BAS 0 B 3      WE       BAS 0 B 1      STOCKS   BAS 0 B 2      LOCO     BAS 0 B 2
SOUNDS    BAS 0 B 3      EXTNDKYB BAS 0 B 4      MESSAGE  BAS 0 B 1      TAPELIST BAS 0 B 2      LOTTO    BAS 0 B 2
SOUNDS2   BAS 0 B 1      EXTNDKYB DOC 1 A 7      OBJECTS  BIN 2 B 2      -------------------     MARTHA   BAS 0 B 1
SWAN      BIN 2 B 1      FREE     BAS 0 B 1      ODIE     PIX 2 B 3                              MAZE     BAS 0 B 2
SYNMUSIC  BIN 2 B 4      GRADBOOK BAS 0 B 1      TANK     BIN 2 B 4      PD-75 Misc. Pgms       MAZE2    BAS 0 B 3
DEEPPURP  BIN 2 B 5      GRNDSTFF BAS 0 B 1      TRIANGLE BIN 2 B 2                              MOTRCYCL BAS 0 B 3
ALFEX     BIN 2 B 2      INSTR    BAS 0 B 1      WORLDMAP BIN 2 B 4      WHODUNIT BAS 0 B 7      PRINTER  BAS 0 B 1
BACH      BIN 2 B 4      LET      BAS 0 B 3      PAINT    BAS 0 B 1      HOUSEADV BAS 0 B 5      SIEVE    BAS 0 B 1
BUMBLE    BIN 2 B 3      STOCKS   BAS 0 B 5      SCRDATA  BIN 2 B 1      VIKINGS  BAS 0 B 9      SLOTS    BAS 0 B 4
CANON     BIN 2 B 3      TWOLINER BAS 0 B 1      RES      BAS 0 B 1      MANSION  BAS 0 B 6      STRIPJAK BAS 0 B 1
DIAMOND   BIN 2 B 3      ATOMS    BAS 0 B 2      SCAN     BAS 0 B 1      HIRESABC BAS 0 B 1      SUPRMIND BAS 0 B 4
ENTAIN    BIN 2 B 1      BEAST    BAS 0 B 1      -------------------     ML TEXT  BIN 2 B 1      TARGET   BAS 0 B 1
FUNERAL   BIN 2 B 3      -------------------                             LORESABC BAS 0 B 1      TEENBDGT BAS 0 B 3
GRENGRSS  BIN 2 B 4                              * PD 72 Basic and                              WUMPUS   BAS 0 B 3
```

**PD-78 Basic Pgms**

| | | | | |
|---|---|---|---|---|
| AUTO LOG | BAS | 0 | B | 2 |
| BOMBER | BAS | 0 | B | 2 |
| CHECKERS | BAS | 0 | B | 2 |
| CLOCK | BAS | 0 | B | 1 |
| COLRCAGE | BAS | 0 | B | 2 |
| DATABASE | BAS | 0 | B | 2 |
| DAY DATE | BAS | 0 | B | 1 |
| DEFEND | BAS | 0 | B | 3 |
| EAGLE | BAS | 0 | B | 1 |
| ENGTOMET | BAS | 0 | B | 6 |
| GRAN-MAP | BAS | 0 | B | 1 |
| ICBM'S | BAS | 0 | B | 3 |
| ILLUSION | BAS | 0 | B | 1 |
| INVADERS | BAS | 0 | B | 2 |
| LIFE | BAS | 0 | B | 2 |
| LPVII | BAS | 0 | B | 1 |
| LTRWRTR | BAS | 0 | B | 3 |
| MATH HLP | BAS | 0 | B | 5 |
| MINIEDIT | BAS | 0 | B | 2 |
| MLADFIND | BAS | 0 | B | 2 |
| ROLL ON | BAS | 0 | B | 2 |
| SCRNPRNT | BAS | 0 | B | 2 |
| SCRPRT | BAS | 0 | B | 1 |
| SUNDANCE | BAS | 0 | B | 2 |
| TASKARTE | BAS | 0 | B | 5 |
| UFO | BAS | 0 | B | 2 |
| WRONGWAY | BAS | 0 | B | 1 |

**PD-79 Misc Pgms**

| | | | | |
|---|---|---|---|---|
| COCO | BAS | 0 | B | 1 |
| COUNT | BAS | 0 | B | 1 |
| AREACODE | BAS | 0 | B | 3 |
| BOMBER | BAS | 0 | B | 2 |
| FINANCE | BAS | 0 | B | 3 |
| FISHPOND | BAS | 0 | B | 4 |
| FONTGEN | BAS | 0 | B | 5 |

| | | | | |
|---|---|---|---|---|
| FONT | DOC | 1 | A | 4 |
| FONT1 | FNT | 1 | A | 1 |
| OUTLINE | FNT | 1 | A | 4 |
| SCRIPT | FNT | 1 | A | 1 |
| SHADOW | FNT | 1 | A | 1 |
| FRUST | BAS | 0 | B | 3 |
| L-CHNGES | BAS | 0 | B | 1 |
| MAS-MIND | BAS | 0 | B | 1 |
| OUTPOST | BAS | 0 | B | 2 |
| P-MAN-4K | BAS | 0 | B | 1 |
| PHONELST | BAS | 0 | B | 2 |
| PINGPONG | BAS | 0 | B | 2 |
| RUINS | BAS | 0 | B | 3 |
| SPIDERS | BAS | 0 | B | 3 |
| TYPING | BAS | 0 | B | 5 |
| USED-TRK | BAS | 0 | B | 1 |
| WORDFIND | BAS | 0 | B | 4 |

**PD-80 Basic Pgms**

| | | | | |
|---|---|---|---|---|
| 3-D-T | BAS | 0 | B | 2 |
| ADRLABEL | BAS | 0 | B | 2 |
| ATTACMAN | BAS | 0 | B | 3 |
| BRICKS | BAS | 0 | B | 1 |
| CANETRAK | BAS | 0 | B | 3 |
| CLINIC | BAS | 0 | B | 7 |
| CONNECT4 | BAS | 0 | B | 2 |
| CONTOUR | BAS | 0 | B | 1 |
| COPTER | BAS | 0 | B | 2 |
| DARTBORD | BAS | 0 | B | 2 |
| DATABASE | BAS | 0 | B | 5 |
| DOS-OR-9 | BAS | 0 | B | 2 |
| ECBDEMO | BAS | 0 | B | 3 |
| GOLDMINE | BAS | 0 | B | 3 |
| IN-OUT | BAS | 0 | B | 1 |
| KALVOS | BAS | 0 | B | 1 |
| MEMBER | DAT | 0 | B | 1 |
| MLADRESS | BAS | 0 | B | 2 |
| MUSIC | BAS | 0 | B | 2 |

| | | | | |
|---|---|---|---|---|
| PATTERN | BAS | 0 | B | 1 |
| PEEK | BAS | 0 | B | 1 |
| PILLBOX | BAS | 0 | B | 2 |
| QUIKDRAW | BAS | 0 | B | 2 |
| SIMON | BAS | 0 | B | 1 |
| SLITHER | BAS | 0 | B | 1 |
| SONGS | BAS | 0 | B | 3 |
| TICTACTO | BAS | 0 | B | 2 |
| TRSCLEAN | BAS | 0 | B | 1 |

**PD-81 Basic Pgms**

| | | | | |
|---|---|---|---|---|
| ALPHABET | BAS | 0 | B | 2 |
| BABYSIT | BAS | 0 | B | 3 |
| BRIDGE | BAS | 0 | B | 8 |
| CLASSIC | BAS | 0 | B | 4 |
| COGS | BAS | 0 | B | 1 |
| FIREWORK | BAS | 0 | B | 2 |
| GIFTWORD | BAS | 0 | B | 4 |
| GO? | BAS | 0 | B | 3 |
| HERALDRY | BAS | 0 | B | 7 |
| LISSAJOS | BAS | 0 | B | 1 |
| MORG&INT | BAS | 0 | B | 1 |
| MOSAICS | BAS | 0 | B | 3 |
| OANDX | BAS | 0 | B | 2 |
| SALE | BAS | 0 | B | 4 |
| SHIFT | BAS | 0 | B | 2 |
| T-T-TEST | BAS | 0 | B | 3 |
| TESSEL | BAS | 0 | B | 3 |
| WORDPRO+ | BAS | 0 | B | 2 |
| YOURGRAF | BAS | 0 | B | 2 |
| 3BALLADS | BAS | 0 | B | 1 |

**PD-82 Misc Pgms**

| | | | | |
|---|---|---|---|---|
| COUNT | BAS | 0 | B | 1 |
| SKPCOUNT | BAS | 0 | B | 1 |
| SPELL | BAS | 0 | B | 1 |

| | | | | |
|---|---|---|---|---|
| READLEVL | BAS | 0 | B | 2 |
| ATBY | BAS | 0 | B | 1 |
| HTO | BAS | 0 | B | 1 |
| ANIMALS | BAS | 0 | B | 1 |
| HANGMAN | BAS | 0 | B | 2 |
| CAMEL | BAS | 0 | B | 1 |
| HAMMURAB | BAS | 0 | B | 2 |
| PUZZLE | BAS | 0 | B | 3 |
| MAZE | BAS | 0 | B | 2 |
| PLACVLU1 | BAS | 0 | B | 1 |
| PLACVLU2 | BAS | 0 | B | 1 |
| LEMONADE | BAS | 0 | B | 5 |
| TCLOCK | BAS | 0 | B | 3 |
| FLASH+ | BAS | 0 | B | 3 |
| FLASHX | BAS | 0 | B | 3 |
| ADDSUBWK | BAS | 0 | B | 3 |
| DAZTERM | BIN | 2 | B | 8 |
| PUZZLE2 | BAS | 0 | B | 3 |
| GAME | BAS | 0 | B | 1 |
| SORCERER | BAS | 0 | B | 6 |

**PD-83 Basic Pgms**

| | | | | |
|---|---|---|---|---|
| KINGTUT | BAS | 0 | B | 7 |
| SURVIVAL | BAS | 0 | B | 5 |
| LLISTER | BAS | 0 | B | 5 |
| CASTLE | BAS | 0 | B | 8 |
| FLASH | BAS | 0 | B | 1 |
| MENU | BAS | 0 | B | 5 |
| SILLY | BAS | 0 | B | 2 |
| STOCKS | BAS | 0 | B | 2 |
| JOUST | BAS | 0 | B | 4 |
| MEMTEST | BAS | 0 | B | 1 |
| CIPHER | BAS | 0 | B | 1 |
| FREEWAY | BAS | 0 | B | 1 |
| GERMS | BAS | 0 | B | 2 |
| OLDHOUSE | BAS | 0 | B | 5 |
| REFLEX | BAS | 0 | B | 1 |
| SUB | BAS | 0 | B | 2 |

| | | | | |
|---|---|---|---|---|
| TANKS | BAS | 0 | B | 2 |
| WATOR | BAS | 0 | B | 3 |

**PD-84 Basic Pgms**

| | | | | |
|---|---|---|---|---|
| LOSTADV | BAS | 0 | B | 4 |
| MTERMFIG | BAS | 0 | B | 3 |
| ORGQUEST | BAS | 0 | B | 2 |
| DRAW | BAS | 0 | B | 2 |
| STOCKS | BAS | 0 | B | 3 |
| QUME | BAS | 0 | B | 1 |
| WATTS | BAS | 0 | B | 1 |
| DIARY | BAS | 0 | B | 2 |
| BRICK | BAS | 0 | B | 2 |
| CLOCK | BAS | 0 | B | 1 |
| UGLYWORD | BAS | 0 | B | 2 |
| PMONE | BAS | 0 | B | 1 |
| PMTWO | BAS | 0 | B | 1 |
| TIMERACE | BAS | 0 | B | 2 |
| VOICE | BAS | 0 | B | 2 |
| PIECHART | BAS | 0 | B | 2 |
| LISTER | BAS | 0 | B | 1 |
| MLTTD | BAS | 0 | B | 1 |
| ALPHA | BAS | 0 | B | 1 |
| RPM | BAS | 0 | B | 1 |
| DIR | BAS | 0 | B | 1 |
| GRADBK | BAS | 0 | B | 4 |
| BRODRIVE | BAS | 0 | B | 1 |
| KEENO | BAS | 0 | B | 2 |
| WATERCST | BAS | 0 | B | 2 |
| TANK | BAS | 0 | B | 2 |
| WND | BAS | 0 | B | 1 |
| UNIFILE | BAS | 0 | B | 3 |
| TIMPIST | BAS | 0 | B | 2 |
| DATAPRNT | BAS | 0 | B | 1 |
| RACES | BAS | 0 | B | 2 |
| SQUEEZE | BAS | 0 | B | 2 |

# PUBLIC DOMAIN SOFTWARE

This large collection of programs will allow you to quickly expand your library. All programs are available on disk and programs with a * are available on tape. Due to the high cost of tapes, programs will be placed on both sides of the tape for orders on 2 or more collections. Some programs require a joystick. Instructions are included in some collections as DAT, DOC, or TXT files. Prices are as follows:

1-4 $4.95, 5-9 $4.50, 10-24 $4, 25 up $3.50

Add $1 shipping for less than 10 and $2 for 10 up.

# KING JAMES BIBLE ON DISK

The complete King James Bible is in short ASCII files for use with word processors such as Telewriter and VIP. These are excellent study sources for Pastors and Bible teachers.

**NEW TESTAMENT** - 8 disks $12

**KJV BIBLE SET** - 34 disks in a 50 disk box $50.

Note: We pay shipping on Bible Orders. Give UPS address for the KJV Bible Set. Foreign orders add shipping.

Checks, VISA & MC Cards

# Dynamic Electronics Inc.

# C PROGRAMMING

## *INCREMENTING VARIABLES*

by
**Norm Matice**

This month we will look at how C handles incrementing variables. We will also look at the C while command because it is part of the program below.

In C we can increment a variable the same way we do in BASIC. We can write a=a+1. This is perfectly legal in C and will get the job done. However it is very rare to see a line incrementing a variable that way in a C program.

The answer to why that is, is very simple. C has a shorthand way to write the same line. The C equivalent to a=a+1 is a++. The variable name followed by two plus signs. In case you are wondering if C will allow you to decrement by using some- thing along the lines of a--.

There is another form of this incrementing that you should know about. That is the ++a form which is a variable preceeded by two plus signs (or two minus signs if decrementing).

The best way to demonstrate the various forms is to do a line by line analysis of the program listed below and explain the results of each section. After that you can type it in, compile the program, and see for yourself how things work.

As always I have started with #include <stdio.h> for my first line. Again this program should run alright without it, but it is a good habit to include it in all programs. The second line is main(), and of course all C programs need main() in them at the start of one function. The third line contains the opening brace, again common to all C functions.

Line 4 is a declaration of variables, in this case only a as an integer. Line 5 is a space for readability. Lines 6 - 8 comprise the first section of the program. Line 6 is the while command. What it is saying, in effect, is that while a is less than 4 carry out the commands in the braces following the while statement. There is no limit (except those imposed by the size of the system) to how many commands we put in the braces following the while command. In this case line 7 prints a message on the screen, so we can see what is going on and line 8 increments the variable a the way we are used to seeing it. When a is no longer less than 4 we will drop out of the while statement and go on.

Line 9 is a space, line 10 just sets a back to 0 for the next section of the program. Line 11 is identical to line 6 and line 12 performs the same function as line 7 did. Line 13 has the new way of incrementing in it. When you run the program you will see that it operates exactly the same as the older form.

Line 14 is a space, line 15 is the while statement. This time though it looks a little different. Now the a++ is up in the while statement. The while statement will allow us to increment inside of it with the a++. What the computer does is check to see if a is less than 4, if it is, it does the statement followwing the while statement. In this case it is another printf so we can see what's happening. What's happened to the braces in this section? If there is only one statement following

a while statement, the computer will use the semi-colon to designate the end of the while statement. If you'll notice the while statement is one of the few in C that is not followed by a semi-colon. That is because the compiler knows that something must follow the while. It will use the end of that something (be it a single line's semi-colon or a group of lines end brace) to designate the end of the while's territory.

The last section again starts with space and a line returning the value of a to 0. This time the while statement has ++a. When we put the ++ in front of the a it tells the computer to increment the a first, then check to see if it is less than 4. When you run the program you will see only three loop 4's on the screen. That is because a is incremented to 1 before the first comparison to 4, therefore there are only three trips through the loop before the condition of being less than 4 is no longer meet and the computer moves on. The last line is of course the closing brace.

loop.c

```
#include <stdio.h>
main()

    int a=0;

    while (a < 4)
     printf("loop 1");
      a=a+1;

    a=0;
    while (a < 4)
     printf("loop 2");
      a++;

    a=0;
    while (a++ < 4)
     printf("loop 3");

    a=0;
    while (++a < 4)
     printf("loop 4");
```

# HAM RADIO & COMPUTERS
## by
## bill chapple w4gqc

I have really enjoyed writing this section and developing software and hardware for Radio Shack Color Computers. Although there is much more that I would have liked to do, there is also much that has been accomplished since this series started.

Due to economics, this will have to be our final edition. I had thought that our subscriptions would increase, but recently we have had a drop in subscriptions forcing us to quit. I will still be developing ham software for computers, and if you are interested in being on my mailing list let me know.

Due to the many requests I have had for a simple Morse code interface, I developed a simple keyer circuit that could be built inside cable connectors. This was presented last Fall. Last month I presented a Morse code decoder that required a simple interface circuit that could also be built inside a cable. This plugs from your transceivers audio output jack to the right joystick port of your computer. An external speaker is required to monitor the audio. For details on building these interfaces refer to issue #58 for the receiver interface and issue #55 for the transmitter keyer interface.

## MORSE TERM 2

This month I combined the two programs to produce my second Morse terminal program. The critical part is the receiver interface. This is connected from joystick 0 to the receiver's audio. This can be checked out by the following basic program:

```
10 A=JOYSTK(0):?A;:GO TO 10
```

As characters are received, the value of A will change. You can adjust the receiver's volume until you notice the value change. This check may not be needed, but can be used just to check out the interface.

The program automatically checks for a color computer 3 and gives the option of operating at double speed or normal speed. Double speed works the best and should be used if your computer can handle the double speed.

The program comes up in the receive mode. I did not include an indicator for the audio level. However I did not find this to be a disadvantage. It will lock up if a signal is not received because it is looking for audio.

Because some people leave too much spacing between character elements, I retained the wide spacing feature from last month. To enable it press the right

arrow key. To return to close spacing, press the left arrow.

To go to transmit, press the "T" key. The transmit menu will then appear. Any of the prepro- grammed messages can be sent by pressing the letter or number next to the message. The mes- sages are set up for my QTH and equipment. These are in the last data statements of the program. They consist of two parts which are the title and the actual data. Quotations must be used if you want commas to be in the data. Basic uses commas to seperate different data elements. Notice how I used commas for my city and town with quotation marks.

To send from the keyboard press the ENTER key and then start sending. To return to the menu press the up arrow. To return to the receive section press the "R" key from the menu.

I tried the terminal program out with very good success. I believe that it works as well or better than the first program I developed with the hardware interface. I even copied some DX stations with it. 73's and I hope these articles have been of benefit to you. - Bill.

```
10 CLEAR 10000
20 PCLEAR1
2000 KK=.5 'NARROW SPACING
2010 DEFUSRO=30031

2020 DIM N$(100),T$(15),D$(15),A
$(130) 'SET UP ARRAY FOR THE CHA
RACTERS
2030 CLS:PRINT"MORSE TERM 2
2040 PRINT"cOPYRITE (c) 1989

2050 PRINT"dYNAMIC eLECTRONICS i
NC.

2060 PRINT"WRITTEN BY BILL CHAPP
LE W4GQC

2070 PRINT:PRINT"PRESS THE RIGHT
 ARROW FOR WIDE  SPACING ON RECE
IVE AND THE LEFT ARROW FOR NARRO
```

```
W SPACING. PRESS THE S KEY TO SH
OW THE SPACING   THAT IS PRESENT
LY IN USE.
2080 GOSUB 3080

2090 IF PEEK(33021)=50 THEN C3=2
 'COLOR COMPUTER 3 TEST

2095 IF C3=2 THEN PALETTE 12,63:
PALETTE 13,0

2100 PRINT"PRESS D FOR DOUBLE SP
EED

2110 D$=INKEY$:IF D$="" THEN 211
0
2120 IF D$="D" THEN DS=1

2130 IF DS=1 THEN POKE 65495+C3,
1 ELSE POKE 65494+C3,0
2140 ' LOADM ML SUB

2150 FOR J=30000 TO 30072:READ A
:POKEJ,A:NEXTJ
2160 'INITIALIZE THE ARRAY

2170 FOR J=0 TO 129:A$(J)=CHR$(3
2):NEXTJ
2180 'DEFINE THE CHARACTERS
2190 A$(5)="A":A$(24)="B"
2200 A$(26)="C":A$(12)="D"
2210 A$(2)="E":A$(18)="F
2220 A$(14)="G":A$(16)="H"
2230 A$(4)="I":A$(23)="J
2240 A$(13)="K":A$(20)="L
2250 A$(7)="M":A$(6)="N
2260 A$(15)="O":A$(22)="P
2270 A$(29)="Q":A$(10)="R
2280 A$(8)="S":A$(3)="T
2290 A$(9)="U":A$(17)="V
2300 A$(11)="W":A$(25)="X
2310 A$(27)="Y":A$(28)="Z
2320 A$(63)="0":A$(47)="1
2330 A$(39)="2":A$(35)="3
2340 A$(33)="4":A$(32)="5
2350 A$(48)="6":A$(56)="7
2360 A$(60)="8":A$(62)="9
2370 A$(85)=".":A$(115)=",
2380 A$(76)="?":A$(97)="*
2385 A$(50)="/"
2390 PRINT

2395 FOR J=1 TO 10:READ T$(J),D$
(J):NEXT J

2400 X=USRO(0) 'GET TIME FROM ML
  SUBROUTINE
```

```
2410 A=PEEK(29999) 'A TELLS WHET
HER WE WERE TIMING DATA OR SPACE
S

2420 IF A=0 THEN 2510 '0 IS A SP
ACE

2430 'THE FOLLOWING ARE AUTOMATI
C SPEED ADJUSTMENTS
2440 IF X>(6*S) THEN S=2*S
2450 IF X<(S/2) THEN S=S/2
2460 IF S<1 THEN S=2
2470 W=0:IF X>=2*S THEN W=1
2480 Q=2*Q+W

2490 IF Q>=128 THEN Q=0 '128 IS
THE LARGEST CHARACTER WE HAVE.
IF GREATER THAN 128 THEN MUST BE
 AN ERROR.
2500 GOTO 2400
2510 P$=INKEY$:IF P$=""THEN 2570

2520 ' THE FOLLOWING ALLOWS FOR
ADDITIONAL SPACING BETWEEN BITS.

2530 IF P$=CHR$(8) THEN KK=.5:GO
TO2570

2540 IF P$=CHR$(9) THEN KK=2.0:G
OTO2570

2550 IF P$="S" AND KK=.5 THEN PR
INT"CLOSE SPACING"

2560 IF P$="S" AND KK=2.0 THEN P
RINT"WIDE SPACING
2564 IF P$="T" THEN 4360
2570 IF X<KK*S THEN 2400

2580 PRINTA$(Q);:IF X>5*S THEN P
RINT" ";

2590 IF JOYSTK(0)<55 THEN 2600
THIS IS SPACE BETWEEN WORDS.

2595 XC$=INKEY$:IF XC$="" THEN 2
590

2597 IF XC$="T" THEN 4360 ELSE 2
590
2600 Q=1
2610 GOTO2400

2620 DATA 189,169,222,182,1,90,1
29,55,,36,3,134,1,57,79,57,18,18

2630 DATA 79,183,117,47,57,18,18
,18,18,18,18,18,18,18,141,223,18
3,117,47,79,95,253,117
```

```
2640 DATA 38,141,213,177,117,47,
38,18,252,117,38,195,0,1,16,131,
255,220,36,6,253,117,38,32

2650 DATA 232,18,252,117,38,126,
180,244,
3080 '

4130 Z=10 'THIS IS THE SPEED VAL
UE

4140 PRINT"PRESS RIGHT ARROW TO
INCREASE":PRINT"SPEED

4150 PRINT"PRESS - KEY TO DECREA
SE SPEED

4160 PRINT"@ KEY SENDS THE BT CH
ARACTER.
4210 'EMPTY THE ARRAY

4220 FOR K=0 TO 99:N$(K)="":NEXT
 K
4230 'DEFINE THE CHARACTERS

4240 N$(8)="IIIIIIII" 'ERROR BAC
K SPACE

4250 N$(46)="IDIDID":N$(44)="DDI
IDD":N$(63)="IIDDII"' . , ?

4260 N$(64)="DIIID":N$(47)="DIID
I" ' BT  /

4270 N$(48)="DDDDD":N$(49)="IDDD
D"' 0 & 1

4280 N$(50)="IIDDD":N$(51)="IIID
D" '2 & 3

4290 N$(52)="IIIID":N$(53)="IIII
I":N$(54)="DIIII":N$(55)="DDIII"
'4,5,6,7

4300 N$(56)="DDDII":N$(57)="DDDD
I":N$(65)="ID":N$(66)="DIII"'8,9
,A,B
4310 'N$(65)=A

4320 N$(67)="DIDI":N$(68)="DII":
N$(69)="I":N$(70)="IIDI":N$(71)=
"DDI":N$(72)="IIII":N$(73)="II"'
C,D,E,F,G,H,I

4330 N$(74)="IDDD":N$(75)="DID":
N$(76)="IDII":N$(77)="DD":N$(78)
="DI":N$(79)="DDD":N$(8)="IDDI"
'J,K,L,M,N,O,P
```

29

```
4340 N$(81)="DDID":N$(82)="IDI":
N$(83)="III":N$(84)="D":N$(85)="
IID":N$(86)="IIID":N$(87)="IDD"'
Q,R,S,T,U,V,W

4350 N$(88)="DIID":N$(89)="DIDD"
:N$(90)="DDII"'X,Y,Z
4355 RETURN
4359 '
4360 GO SUB 4380
4370 GO TO 4360

4380 CLS:FOR Q=1 TO 9:PRINTQ;T$(
Q):NEXT
4390 PRINT" A "D$(11)
4400 PRINT" B "D$(12)

4410 PRINT" C ";"ENTER STATIONS
CALL SIGN"
4420 PRINT" D ENTER HIS NAME"

4430 PRINT"PRESS UP ARROW TO RET
URN HERE.

4440 PRINT"ENTER FOR KEYBOARD, R
 TO RCV."
4450 Q$=INKEY$:IF Q$=""THEN 4450

4460 IF Q$="C" THEN INPUT "ENTER
 STATIONS CALL SIGN";D$(11):GOTO
4380

4470 IF Q$="A" THEN Q=11:GOTO451
0

4480 IF Q$="D"THEN INPUT"HIS NAM
E";D$(12):GOTO4380
4490 IF Q$="B"THEN Q=12:GOTO4510

4495 IF Q$="R" THEN 2400 'GO TO
RECEIVE SECTION
4500 Q=VAL(Q$)
4510 IF Q=0 THEN 4530

4520 LL=LEN(D$(Q)):FOR JJ=1 TO L
L:P$=MID$(D$(Q),JJ,1):GOSUB 4560
:NEXTJJ:GOTO4380
4530 GOSUB 4540:GOTO4530

4540 P$=INKEY$:IF P$="" THEN 454
0
4550 IF P$=CHR$(94) THEN 4360
4560 IF P$=CHR$(10) THEN 4810

4570 XZ$=INKEY$:IF XZ$<>""THEN 4
360
4580 'IF KEY IS - DECREASE SPEED
4590 P=ASC(P$)

4600 IF P=45THEN Z=Z+1:PRINT"SPE
ED="Z;:GO TO 4540

4610 'INCREASE SPEED FOR RIGHT A
RROW P=9

4620 IF P=9 THEN Z=Z-1:PRINT"SPE
ED ="Z;:GO TO 4540
4630 IF Z=0 THEN Z=1
4640 IF P=32 THEN PRINT" ";
4650 PRINTP$;
4660 P=ASC(P$):IF P=8 THEN 4680
4670 'IF P<33 THEN 430
4680 N=P

4690 IF N$(N)="" THEN PRINTCHR$(
8);:RETURN
4700 L=LEN(N$(N))
4710 '
4720 'THIS DECODES THE CHARACTER

4730 FOR J=1 TO L:X$=MID$(N$(N),
J,1):IF X$="D" THEN Y=3 ELSE IF
X$="I" THEN Y=1
4740 W=Y*Z
4750 IF W<1 THEN W=1
4760 'SEND DOT OR DASH

4770 POKE65312,0:FOR PP=1 TO 4*W
:NEXT PP:POKE65312,2:FOR PP=1 TO
 Z:NEXT
4780 NEXT J
4790 RETURN

4800 'THIS PRINTS COMMENTS ON SC
REEN
4820 IF X$=CHR$(10) THEN 4540
4830 GOTO 4810
4840 '

4850 DATA SEND CQ,CQ CQ CQ DE W4
GQC W4GQC W4GQC CQ CQ CQ CQ DE W
4GQC W4GQC W4GQC K

4860 DATA FIRST TRANSMISSION," T
NX FOR THE CALL OM @ THE NAME HE
RE IS BILL BILL QTH IS HARTSELLE
,AL HARTSELLE, AL @"
4870 DATA DE W4GQC K,DE W4GQC K

4880 DATA SECOND TRANSMISSION, T
HE RIG HERE IS A YAESU FT 757GX
WITH A DIPOLE ANTENNA AND A HEAT
H SB200 LINEAR @
4890 DATA W4GQC,W4GQC

4900 DATA CQ DX, CQ DX CQ DX CQ
DX DE W4GQC W4GQC CQ DX CQ DX CQ
 DX DE W4GQC W4GQC W4GQC DX K
4910 DATA ,,,,,,,,,,,,,,,,,,
9000 A=JOYSTK(0):PRINTA;
9010 GOTO9000
```

# PRODUCT REVIEWS

## KING'S QUEST III

### Reviewed by Norm Matice

### Requires 512K CoCo 3

King's Quest III (To Heir is Human) is a 3-D animated adventure game. That description technically defines the game, but in no way does justice to the beauty and intricacies of this program.

King's Quest III starts with the adventurer assuming the identity of the evil wizard Manannan's slave. You were kidnapped as an infant by the wizard and raised to serve him. The wizard likes you ignorant, that way you pose no threat to him. If he finds that you have been able to gain any knowledge he will zap you to dust.

Certainly you want to get your character out of the intolerable situation you found him in. The adventure lies in doing that without the wizard destroying you. You will be able to use items of magic and create magic spells as you play the game. Keep in mind that your only safe haven is your room.

Even if you are able to defeat the wizard, there is still much to be done. Despite the simple description I have given here, there are many hours of adventuring in this game, but I'd rather not give away the plot of the adventure.

The game itself needs a 512K Color Computer 3 to run. If you need a reason to upgrade this could be it. The game uses the 16 color hi-resolution screen. I saw the same game played on an IBM clone with an EGA card (a graphics card that usually cost more than a CoCo 3) and could discern no difference in the screens. A small explanation about the 3-D aspect of the game is needed. 3-D doesn't mean that you will be sitting in front of the screen with a pair of cardboard glasses on, it indicates that the character passes in front of some objects and behind others depending on his location in relation to theirs on the screen. It is a very nice effect.

The game can be played with joysticks or with the keyboard, and can be viewed on a TV, composite or RGB monitor. You will need at least one disk drive. The program can be transferred to a hard drive, which is preferrable because the program covers 5 disks both front and back. I played it on a floppy system and had to switch disks, but that detracted very little from the game.

Let me give one last word of advice. While you are playing the game examine everything carefully. Don't just look at things, look on top of them, under them, behind them, whatever you can think of. The pictures are a delight and the plot is intriguing. If you have 512K or are thinking of upgrading, this game is a must.

It's available at Radio Shack stores and cost $34.95. The Tandy 1000 version is $39.95.

## PHANTOMGRAPH

### Reviewed by Norm Matice

### Requires 512K CoCo 3

Phantomgraph is a program that will transform your raw data into graphs. The program can create four types of graphs, point, vector, bar or pie. The program is menu driven and works quite nicely with a mouse. It will also work with a joystick.

To enter data you simply select the graph option with your mouse. A pull down menu

will appear from which you can select the data option. A box will appear that has a place for entering the value of your data, a label for your data (if you so desire), the color of your data, and a pattern for when the data is drawn on the screen.

Inserting the values you wish to use is done by moving the on screen arrow to the correct spot, pushing a button to eliminate any old data that might be there and then entering your new data from the keyboard. The left arrow key functions as a backspace and the enter key lets you indicate when you are finished.

Other options allow you to label your graph, the x and y axis of your graph, or the sections of your chart. The graph can of course be sent to a printer for a hardcopy version of your graph. The results are good enough to be included in reports you might have to make.

The instruction book that comes with the program contains a few examples of making graphs. It goes through them step by step guiding you through which keys to press and how to make use of the options. The book also indicates that you can run a Dynacalc file into Phantomgraph and print out a graph from it.

Phantomgraph is designed to run on a Color Computer 3 with OS-9 level II. The program needs 512K to run. Its available at Radio Shhack stores and costs $39.95.

## SPECTACULATOR

### Reviewed by Norm Matice

Spectaculator is a ROM-PAC based spreadsheet for the CoCo 1,2 and 3. By now I'm sure that almost everyone knows what a spreadsheet is, but just in case I will give a breif description. A spreadsheet program will allow the user to enter numeric data into a matrix of rows and columns. Each intersection of a row and a column is an element called a cell. These cells can hold a numerical value, a piece of text, or a formula. It is the cell's ability to hold a mathematical formula that makes spreadsheets popular and powerful programs.

If after you have entered all of your data and your formulas, then you can change a value in a row or column and the spreadsheet will recalculate the values for you. This gives you the advantadge of trying several different scenarios with very little effort, after the initial setup.

Back to Spectaculator. Because the program is a ROM-PAC, storage of your spreadsheets is on a cassette tape. If you do not have a disk drive, and you want a spreadsheet, Spectaculator is one of the few programs that outputs to cassette. Certainly if you own a disk drive you will want to consider more powerful spreadsheet programs like Dynacalc.

Spectaculator being written to work on all the Color Computers has the limited 32 column screen. This limits the amount of data that can be viewed on the screen at one time. The program itself does all the spreadsheet basics. It will allow the entering of data, text and formulas. It will calculate by row or column. It will also allow you to output your spreadsheet to either the aforementioned tape or a printer for hardcopy.

Probably the thing Spectaculator has going for it is the price. For $29.95 you can test drive this type of program and see if it is the kind of thing that will fit your needs. Also according to the Radio Shack Software catalog there is a disk version available for $39.95 that will save to disk. Spectaculator is available at Radio Shack stores.

## THE MATH TUTOR

### Reviewed by Doug Canfield

New in Radio Shack's Educa-

tional line of ROM-PACKS is The Math Tutor. The Math Tutor is designed to allow drilling in the areas of arithmetic, fractions and simple Algebra.

When the cartridge is plugeed in, and the computer is turned on, you are faced with a medium resolution graphic display that depicts a shooting gallery at a carnival. You will also hear some typical carnival music. The program is entirely menu driven, and you have no need of the manual unless you want to save or load some "Homework" from tape.

The arithmetic option allows you do problems ranging in difficulty from two single digit positive numbers to two double digit mixed positive and negative numbers. You can add, subtract, multiply, divide, or compare two numbers.

The fractions it can give you range in difficulty from always common denonminator, with no reducing, to not common denominator with reducing. With fractions, you can add, subtract, or compare in value.

You are given a problem like: 20-30+X=-5 in the Algebra mode. You must give the value of X.

In all of the lessons, after doing a predetermined number of problems, you can go to the "Math gallery", which simulates a shooting gallery. A balloon is shown floating across the screen, and if you answer the problem correctly, before it makes it across the screen, then you "shoot" the ballon.

Amost everything can be set as desired in the "controls" option, to allow you to tailor the difficulty to your own abilty.

The Math Tutor works on all Color computers, and can be obtained from Radio Shack for $24.95

## TETRIS

Reviewed by Doug Canfield

One of Radio Shack's new ROM-PACK releases, Tetris, offers a new idea in gaming thought. Instead of matching your reflexes against the onslaught of an evil enemy, or matching your wits against a perilous adventure, you need both your reflexes and your wits to put together a puzzle which is sent to you piecemeal by the computer.

```
                        0
                        0
   00 000  00  0  00  000  0
   00 0    00  000  00    0  0

      X           X
      X      0    X
      X      0    X
      X      00   X
      XO          X
      XO          X
      XO 000    00X
      X000000   00X
      XXXXXXXXXXXX
```

The computer randomly picks a piece shaped like one of the pieces shown above, and drops it into a box which is 10 blocks wide and 20 blocks tall. As the pieces are falling, you have time to rotate them, or move them side to side, until they land on the top of the pile. If the piece completes an unbroken line from one side of the box to the other (as the "L" shaped piece in the picture above will), then that line will disappear, and the rest of the pile will drop down to fill its place. Your score is based on how many lines you are able to complete before the box becomes filled, and on how many pieces have been dropped.

Tetris requires fast reflexes, and quick thinking, to place all of the pieces without getting them jumbled up. To add more challenge, the game speeds up as more lines are completed.

Tetris works fine on a Coco 1 or 2; and, if you use a Coco 3, it gives a better display by using the high-res screens. The game is available for $29.95 at Radio Shack.

# editor's comments

It doesn't seem like five years have passed since I started Dynamic Color News as a monthly newsletter. Our first issue was just a few pages, and it has grown to a 40-50 page monthly magazine. Due to economics this will be our last issue. We never did have a large number of subscribers, and recently we have had a reduction in subscriptions. The magazine has never supported itself, and we have had to supplement it with sales from other products.

However we have had very good support and I want to thank everyone who contributed. Especially I want to thank John Galus for his series on Assembly Language Programming and the Color Computer 3. Norm Matice picked up our OS9 and Basic 09 series last year, recently started the C programming series, and has helped us with product reviews. Doug Canfield has come by and helped us on many occassions with programming and product reviews. He recently agreed to take over our assembly language series. Bob Helms who is in the Air Force contributed several articles. Tandy Corporation supplied us with Radio Shack Products to review. Bill Bernico and T & D Subscription Software supplied us with programs. There were many others who helped us with letters, comments, and programs. I want to say thanks to each of you for your cooperation and support.

We will still be supporting Radio Shack Color Computers. Most of my experience as an engineer has been in hardware design. I plan to use my experience and design new products. I am very much interested in music and am enrolled in music at the University of Alabama in Huntsville.

We are reducing the prices of our back issues and our DCN on disk. If you want to expand your library you can order these. I have a set of our DCN on disk in one disk box. Since editorials were also included in these, I use them to refer back to articles we previously printed by loading them into my word processor. The cumulative index is included in this month's DCN on disk so you will know which disk contains a particular article. Most of our issues are on disk but a few were lost. We will give you a credit of $1.50 a month for each issue you did not receive on your subscription. Look on your address label for the expiration date of your subscription. For example if your expiration date is 10/89 then you have 7 months * 1.50 or $10.50 credit that you can use for anything we have. If you want a cash refund for the balance of your subscription, then we will refund your money as funds become available.

I will still be available if you have a question or problem. You can call in the evenings or on the weekends. I want to again thank each of you for your support and hope Dynamic Color News has been beneficial to you.

# TYPING TUTOR

This program will help you increase your typing speed by giving drills on Home Keys, Home Row, Top Row, Third Finger, Top Row, Pointer Finger, Little Finger, and Bottom Row. After you select the area in which you want practice, words are placed on the screen and you type them. After several words, your speed is indicated on the screen and you can select additional exercises.

```
10 DATA 15,FAD,A,AS,DAD,AD,SAD,L
AD,FALL,ALFALFA,SASS,LASS,DADS,L
ADS,FALLS,FADS

20 DATA 16,HAD,HAS,GAS,SAG,HALL,
HALLS,LADS,SAGS,HAG,LAG,LAGS,SLA
G,SHALL,SASH,DASH,FLASH

30 DATA 17,DEAF,DEED,SEED,FEED,H
EED,LIKE,KILL,FILL,FEEL,FEES,LIE
D,DIAL,SLIDE,FLIES,SLID,LIKE,GLI
DE

40 DATA 17,TAG,HAT,TALL,THY,DAY,
HAY,JAY,GAY,LAY,TAR,RAT,STAR,STA
FF,FAST,TRY,SAY,YARD

50 DATA 20,WISH,EXAM,EXACT,TEXT,
TWO,WON,SOW,WASH,WORSE,OWE,WORD,
LOOK,LOSE,SOD,WOW,TOW,TEXAS,OXEN
,MIX,WORLD

60 DATA 16,QUAKE,QUIZ,QUIP,ZAP,Q
UIT,PIQUE,PLAQUE,PUZZLE,PLAZA,SA
P,ZIPPER,PRIZE,QUICK,SQUEEZE,SEA
L,ZIP

70 DATA 18,CALM,CAN,MEN,NIMBLE,E
XACT,EXAM,MIX,NIX,BUZZ,ZOOM,NAVY
,CAB,BACK,BOMB,ZOMBIE,CAVE,VACAT
E,VARMINT
80 GOSUB 1110
```

```
90 DIM A$(7,20)
100 RESTORE
110 FOR I=1 TO 7
120 READ N
130 FOR J=1 TO N
140 READ A$(I,J)
150 NEXT J
160 NEXT I
170 GOSUB 550
180 GOSUB 740
190 S=0
200 T2=0
210 L2=0
220 FOR K=1 TO 10
230 J=RND(N)
240 IF A$(I,J)="" THEN 230
250 M=465-32*K
260 SOUND 159,3
270 PRINT @ M,A$(I,J)
280 TIMER=0
290 INPUT B$
300 NT=TIMER
310 IF B$=A$(I,J) THEN 340
320 SOUND 5,20
330 GOTO 360
340 PLAY "L16;C;D;E;F;G"
350 S=S+1
360 T2=T2+NT
370 L2=L2+LEN(B$)
380 A$(I,J)=""
390 PRINT @ M,"          "
400 NEXT K
410 FOR X=1 TO 1000
420 NEXT X
430 CLS
440 PRINT @ 131,"YOUR SCORE IS"
450 PRINT @ 196,10*S;"PERCENT AC
CURACY"
460 WORDS=L2/5+2
470 WPM=INT(WORDS*3600/T2+.5)
480 PRINT @ 228,WPM;"WORDS PER M
INUTE"
490 PRINT @ 419,"PRESS C TO CONT
INUE"
```

```
500 C=2
510 GOSUB 1330
520 D$=INKEY$
530 IF D$<>"C" THEN 520
540 GOTO 100
550 CLS
560 PRINT @ 67,"CHOOSE ONE"
570 PRINT @ 131,"1 HOME KEYS"
580 PRINT @ 163,"2 HOME ROW"

590 PRINT @ 195,"3 TOP ROW, THIR
D FINGER"

600 PRINT @ 227,"4 TOP ROW, POIN
TER FINGER"
610 PRINT @ 259,"5 RING FINGER"

620 PRINT @ 291,"6 LITTLE FINGER
"
630 PRINT @ 323,"7 BOTTOM ROW"
640 PRINT @ 355,"8 END PROGRAM"
650 C=3
660 GOSUB 1330
670 C$=INKEY$
680 IF C$="" THEN 670
690 IF ASC(C$)>56 THEN 670
700 IF ASC(C$)<49 THEN 670
710 IF VAL(C$)=8 THEN 1440
720 I=VAL(C$)
730 RETURN
740 CLS
750 SCREEN 0,0

760 PRINT @ 256,"TYPE & ENTER TH
E WORDS YOU SEE."
770 FOR X=1 TO 1000
780 NEXT X
790 CLS
800 C=4
810 Z=17
820 FOR X=34 TO 41
830 FOR Y=Z TO 31
840 SET (X,Y,C)
850 NEXT Y
860 Z=Z-1
870 NEXT X
880 Z=12
890 FOR X=42 TO 47
900 FOR Y=Z TO 31
910 SET (X,Y,C)
920 NEXT Y
930 Z=Z+1
940 NEXT X
950 FOR Y=26 TO 28
960 SET (33,Y,C)
970 SET (48,Y,C)
980 SET (32,Y+1,C)
990 SET (49,Y+1,C)
1000 SET (31,Y+2,C)
1010 SET (50,Y+2,C)
1020 NEXT Y
1030 FOR Y=10 TO 0 STEP -2
1040 SET (41,Y,2)
1050 SET (40,Y,2)
1060 NEXT Y
1070 PRINT @ 18,"100%"
1080 FOR X=1 TO 600
1090 NEXT X
1100 RETURN
1110 CLS
1120 SCREEN 0,1
1130 PRINT @ 139,"T Y P I N G"
1140 PRINT @ 207,"FOR"
1150 PRINT @ 268,"ACCURACY"
1160 PRINT @ 363,"QWERTYUIOP"
1170 PRINT @ 395,"ASDFGHJKL;"
1180 PRINT @ 428,"ZXCVBNM,."
1190 FOR X=16 TO 46
1200 SET (X,20,3)
1210 SET (X,29,3)
1220 NEXT X
1230 FOR Y=21 TO 28
1240 SET (16,Y,3)
1250 SET (46,Y,3)
1260 NEXT Y
1270 C=4
1280 GOSUB 1330

1290 PLAY "L8;CCE;O1;EGG;L4;E;L8
;F;O3;FDD;O1;BB;L4;G"

1300 PLAY "L8;C;O2;CEEGG;L4;E;L8
;C;O3;C;F#;O2;F#;L4;G;L2;G"
1310 SCREEN 0,0
1320 RETURN
1330 FOR X=0 TO 63
1340 SET (X,0,C)
1350 SET (X,31,C)
1360 NEXT X
1370 FOR Y=1 TO 30
1380 SET (1,Y,C)
1390 SET (0,Y,C)
1400 SET (63,Y,C)
1410 SET (62,Y,C)
1420 NEXT Y
1430 RETURN
1440 CLS
1450 END
```

## Dynamic Color News Discontinued

Dynamic Color News was printed for 5 years. It started as a newsletter and expanded to a 40-50 page magazine. Our last issue was #59 March 1989. Back issues are available in printed form or on disk. These are ex-cellent references for various color computer subjects such as Basic Programming, C, OS-9, Basic 09, Assembly Language, Ham Radio, and hardware. Also basic and machine language programs are included. All of our edi-torials were put on the disk as ASCII files which can be read with any word processor. One disk box can hold all of our disks. The last issue has our cummulative index of all the articles and programs. We printed 59 issues but do not have disks for 5 of the earlier issues. We are offering the complete 54 disk set in a box for $125. You can not buy color computer programs for less than this.

Prices are as follows:

| each | 10 up | Complete Set | |
|------|-------|---------|---|
| $1.50 | $1.25 | $60. | Magazines |
| $4.95 | $4.00 | $150. | Disks |

Magazine sample $2.00 ppd.
Disk sample $5.50 ppd.
Sample of both $7.00 ppd.

Add $3 shipping per order.
Give street address for UPS.

Checks , VISA or MC Cards

**DYNAMIC ELECTRONICS INC**
Box 896 (205) 773-2758
Hartselle, AL 35640

## NEW PRODUCTS

The CHURCH FILE Package was developed over a five year period of actual use in a small church (74 families). The main program is a database that will print out mailings, phone lists, sorted groups within the church, and a church directory. In addition, there is a program to record Church Library holdings, another database for a Sunday School, and one for a Summer Camp. Also included is a Personal Mailing list database. Requires 32K Coco ECB, tape or disk. Price $12.
   Bob Silver - Rt 2 box 24C - Nine Mile Falls, WA 99026

## BULLETIN BOARDS
## COMPUTER CLUBS

If you want a free listing send us the information. These listings will be kept current. Please let us know of any errors or discontinued bulletin boards or clubs.

## BULLETIN BOARDS

Come and enter the realm of the Dungeon BBS of Newport NC. Supporting 300/1200/2400 Baud, Online 24 Hours. The board features Hi-Res graphics, forums, special Interest Groups, Multi Message Bases, News amd Information, Market and Advertising and Upload and Download areas for CoCo IBM, Atari, Apple, Tandy, and Macintosh. 30 MEG Capability. Call 919-726-9737.
Parms: 8-N-1. Chuck Katsekes
410 Scott Dr
Newport, NC 28570

The COCONET Olkahoma City, Ok. area. COCO Net 2. ANSI Terminal. Xmodem downloads & uploads. 300, 1200, 2400 baud.
(405) 751 - 5359

The Computer Cellar, 300/1200/2400 Baud 24 hrs. /day (319) 277-0646 Kevin Langenwalter
2605 Eddie St
Cedar Falls, IA 50613

The CoCo Nut Tree, 300/1200 Baud, 24 hrs, 7 days a week, no parity, 7 data bits, 1 stop bits, echo off. (216) 530-6809

The ~Manton Modem~ BBS 300-1200-2400 Baud, 24 hrs daily, 7 days. PBBS V6.0. Hundreds of CoCo & IBM downloads. (616) 824-6026
SYSOP- Carl Johnson
6030 N. 43rd
Manton, MI 49663

The ~Matrix~ BBS located in Miami Florida was established on OCT 1, 1987. ~Matrix~ runs under OS-9 and is multi-user. 40 megs on-line with Uploads/Downloads, Private Mail, Public Msg Bases, ON-LINE Games, Programers Talk Guild ... etc. 300/1200 Baud, 8-N-1
24 Hrs/ Day, SYOP: Chris Malcolm, (305) 895-2312 -- (305) 899-8506

UTOPIA NETWORK BBS (614) 622- 4061
145 1/2 Park Ave., Coshocton, OH 43812
Hours 7pm-6am 7 days.

COCORAMA (312) 307-1519 7 days, 24 hrs 300-1200 Baud

MINDMASTERS DOMAIN (312) 463-8932 7 days, 24 hrs 300-1200 Baud

FAT NEWSLETTER 300/1200 Baud, FAT 1010
Concord Av Piqua, OH 45356 (513) 778-9624

Hot CoCo Users Group, La Porte City,IA 50651
Monthly News Letter & BBS (319) 277-0646

Electronic Info. Network George Matyaszek
Syslink (312) 622-4442 Chicago Syslink Network 1718 N. Long Av Chicago, IL 60639-4321

## COLOR COMPUTER
## CLUBS

Jan Colucci Editor CS3TCrier The Color Computer Club, PO Box 478, Canfield, OH 44406

Jack Eizenga, Treasurer & Disk Librarian Color America 3811 N. Foster Av, Baldwin Park, CA 91706

Mid Iowa CoCo, Terry Simons,Treas.1328 48th, Des Moines, IA 50311 (515) 279-2576

T-BUG Newsletter, Linda Hapner, 3329 B. Beacon #50, N. Chicago, IL 60064

Glenside Color Com. Club, Serving Chicago & the Western Suburbs, Ed Hathaway (312) 462-0694

Cook Cty Color Computer Club, Serving Chicago and the Southern Sub. Tony Nowakowski (312) 895-0393

Illinois C.C. Club of Elgin, Serving the Northwest Suburbs of Chicago, Tony Podraza. (312) 428-3576

Greater Lansing Color Comp Users Group Dale Knepper, Pres., PO Box 14114, Lansing, MI 48901. Meets every 4th Tuesday at 7:00pm

Benchboard BBS Online Since 1984! 300/1200/ 2400 Baud, 8-N-1 24 Hrs/Day, 40 Megs online! Charter member of the Modem Support Group. (517) 394-2447



*ADVERTISER'S INDEX*

We would appreciate it if you would let these advertisers know that you saw their advertisement in Dynamic Color News

**#35 Feb 87**
ML Programming (10)
Interfacing Comput
(Joystick Voltmeter)
Ham Radio & Com. (7)
Scrolling Around
Basic Programming
Address File

**#36 Mar 87**
OWARE (Game)
Joystick Ohmmeter
EPROMS (1)
ML Programming (11)
CoCo 3 (Lowercase
  Char & Rev.Video)
Ham Radio & Com.(8)
Basic Programming
Invoice Program

**#37 Apr 87**
Diver (Game)
EPROMS (2)
ML Programming (12)
Using Joystick Port
for Measuring Temp.
Ham Radio & Com.(9)
CoCo 3 (Error Trap)
Basic Programming

**#38 May 87**
Joystick Digital
  Thermometer
Accounts Payable
 (Business Pgm)
EPROMS (3)
ML Programming (13)
CoCo 3 Mem. Manager
Ham Radio & Com.(10)
Basic Programming
 (Fast Sorting)

**#39 June/July 87**
Geneology Program
ML Programming (14)
Measurinng Light with
  Joystick Ports
EPROMS (4)
CoCo 3 Graph. Demo
Ham Radio & Com.(11)
Calendar Program

**#40 Aug 87**
Job Costing Program
ML Programming (15)
Basic Programming
Ham Radio & Com.(12)
Compound Interest Pgm
CC-3 Hi-Res Graphics
  Save Program
Dog Race Program

**#41 Sep 87**
Astro Dodge Game
ML Programming (16)
Reformatting Data(1)
Meteors Program
Computer Terminology
Ham Radio & Com.(13)
Relay Interface
 (Hardware Project)

**#42 Oct 87**
Taking Control (1)
 (Basic Programming)
ML Programming (17)
Disk Cataloger Pgm

Reformatting Data(2)
Parachute (Game)
Ham Radio & com.(14)
HAM RTTY Program

**#43 Nov 87**
Save the Maiden (G)
Taking Control (2)
ML Programming (18)
Reformatting Data(2)
Music Program
HAM Radio & Com.(15)
 (Packet Radio)

**#44 Dec 87**
Kingpede (Game)
Taking Control (3)
Printer Utilities
Ham Radio & Com.(19)
Audio Generator Pgm
Reformatting Data(3)
Screen Dump Program

**#45 Jan 88**
Living Maze (Game)
ML Programming (20)
Ham Radio & Comp.(17)
Frequency Counter Pgm
Taking Control (4)
Reformatting Data (6)
FANTASY2 Music Pgm

**#46 Feb 88**
Using ROM Routines
Taking Control (5)
OS-9 (New Series)
BARRACAT (Card Game)
Ham Radio & Com (18)
Improved Ham Log Pgm

**#47 Mar 88**
Using ROM Rout. (2)
Taking Control (6)
OS-9 & Basic O9
Coast-Coast (Game)
Logic (Ed. Game)
Forth Prog. Lang.
ML Loader Pgm.
Ham Radio & Com (18)
WEFAX -Weather Pgm.

**#48 Apr 88**
Marriage of ML & Bas
Area Code (Program)
OS-9 & Basic O9
Taking Control (7)
 (Basic Programming)
Forth Progamming (2)
Fast Dir (Disk Pgm)
Ham Radio & Com.
Tuning Meter Pgm
Atlanta (Picture)

**#49 May 88**
Advance (Game)
Marriage of ML & Bas
Superspell (Ed. Pgm.)
OS-9 & Basic O9
Sink the Ships (Game)
Basketball Math (G)
Taking Control (8)
Ham Radio & Comp.
Ham Math Program

**#50 June 88**
OS-9 & Basic O9

Media Master (Pgm)
Mysterious Island (G)
Taking Control (9)
  Basic Prog.
Improved Word Proc.
Forth & Forth-Editor
Ham Radio & Comp.
Improved RTTY Program
Airplane (Game)

**#51 July 88**
OS-9 & Basic O9
Using Interrupts
Maze (ML Game)
Forth Background
Taking Control (10)
Player Guitar (prog)
Kwik Kopyur (Disk UT)
Ham Radio & Comp.
Audio Squelch Prog
Hardware
Motor Jump (game)
Bustout (game)

**#52 Aug 88**
OS-9 & Basic O9
Trio (three Progs)
COCO III (new series)
HI-Res Screen Saving
  With Program
EDT-MAC (AL Ut)
Taking Control (11)
Teacher Grade Prog
Ham Radio & Comp
Handy Ham Prog
Hardware
 (RS-232 switch)
U.S. States (prog)
Loan Program

**#53 Sept 88**
OS-9 & Basic O9
COCO III (part 2)
Educational TRIO
Neighboring States
Basic Programming
(Direct Access Files)
Direct Access File Pgm
Aliens (CoCo 3 game)
Ham Radio & Comp
Bible Quiz

**#54 Oct 88**
OS-9 & Basic O9
COCO III (part 3)
Squares (game)
CoCo COOTIE (game)
Basic Programming
(Using Disk Files)
Grade Program
Flashcard (Ed. Pgm)
Ping Pong (game)
Ham Radio & Comp
Improved Morse
  Keyer Pgm
World Map (Pgm)

**#55 Nov 88**
OS-9 & Basic O9
(Redirection)
COCO III (part 4)
Super Blitz (game)
Disk Drive Problems
(Double side util.)
Ham Radio & Comp
Morse Keyer Interface
Forward (game)

Basic Programming
 (Disk I/O)
Disk I/O Program
Drill (Math Pgm)

**#56 Dec 88**
OS-9 & Basic O9
(Using Multiuser Feat)
Screen Dump Pgm
( DMP-105 printer)
COCO III (part 5)
Assembly Language
Basic Prog.-DSK I/O
ML Address Finder Pgm
Ham Radio & Computers
SWR & Power Pgm
Hangman (word game)

**#57 Jan 89**
Assembly Language
COCO III (high
  resolution Tx)
Air Attack  (game)
Basic Programming
(developing a ML Loader)
Machine Language Loader
C Programming (new)
Math (educational game)
Ham Radio & Computers
VHF Antenna Design Pgm
AUTOEXEC Pgm
OS-9 & Basic O9

**#58 Feb 89**
OS-9 &Basic O9
Tilertex (game)
COCO III (part 7)
COCO III Color
      Graph (pgm)
Assembly Language
C Programming
Basic Programming
ASCII Basic Pgm
      Restorer
Robots (game)
Ham Radio & Com.
Morse Cde. Decoder (pgm)

**#59 March 89 (Final Issue)**
OS-9 &Basic O9
Taxman (game)
Assembly Language
Character Generator (pgm)
Basic Programming
Word Processor Pgm
C Programming
Ham Radio & Computers
Morse Term 2
Typing Tutor (pgm)


Prices as follows:

|        |          | complete |
|--------|----------|----------|
| each   | 10 up    | set      |

**Magazines**

| $1.50 | $1.25 | $60 |

**Disks**

| $4.95 | $4.00 | $150 |

Add $3 shipping
Give Street Address
 for UPS.

# CLASSIFIED ADS

# DYNAMIC ELECTRONICS INC.
## P. O. Box 896    (205) 773-2758
## Hartselle, AL  35640