ENGINEERING NOTES
on
Radio Shack Color Computers

September 1985
Vol. 2 No. 8

$1.95

SYNTAX ERROR

COLOR NEWS

* FAST FOOD PROGRAM
* BAR GRAPH PROGRAM
* VECTOR CORRECTOR PROGRAM

DYNAMIC COLOR NEWS is published monthly by DYNAMIC ELECTRONICS, INC., P.O. Box 896, Hartselle, AL 35640, phone (205) 773-2758. Bill Chapple, President; Alene Chapple, Sec. & Treas.; John Pearson, Ph. D. Consultant; Bob Morgan, Ph. D., Consultant.

 Subscriptions are $15/yr for U.S.A. & Canada, $30 other foreign.

The purpose of this newsletter 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 newsletter are encouraged. All submissions become the property of Dynamic Electronics 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 newsletter. 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 BIN file.

****
*
* DYNAMIC COLOR NEWS *
*
* September 1985 *
*
* Editor and Publisher *
* Bill Chapple *
*
* Secretary *
* Deanne Hill *
*
****

CONTENTS

EDITOR'S COMMENTS

How much memory do you need? There seems to be the idea that the more memory one has the better. In a way this is true. Obviously if you don't have enough memory you can't do much. The first color computers had only 4K of memory with about 2K being available for programs. Now 16K is standard and 64K is easy to obtain.

A 64K computer is very good especially if you have a disk drive which allows programs to be quickly loaded and saved. We also use tape some although it is much slower. Larger memories are good for ramdisks or for storing large files and programs. Most programs are designed for 32K maximum although there are a few that require 64K. You can do a lot with an 8K or 16K program. The purpose of this newsletter is to show what can be done by giving instruction and examples. We have given short programs that do big jobs. Most of us don't begin to tap the programming ability of our computers. For example look at the programs that we included within these issues. They are not very long but do some very impressive tasks.

We started the series on writing programs to teach our readers how to write efficient and effective prorams. We have covered some very useful concepts in this series and can write powerful programs. As an example look at the "FAST FOOD" program in this issue. I am sure that everyone of our readers has placed an order for food and had it entered on a computer. Our program could be used for such an order taking task. All of the commands used in this program were covered in our editorials.

What about purchased software? About 15 years ago before I even cared about computers a friend of mine said "Bill get you a computer and you can program it to do whatever you want". This is very true. My computer can be used for figuring income tax, check books, sending Morse code, displaying messages, playing games, or controlling devices. It can even talk or recognize voice commands. I can change what it does by loading a different program. Software runs from about $10 to $60 more or less. It is not very expensive to purchase a package to allow your computer to do a different task. I have purchased several for my own use and have not been disappointed.

What about future computer expansions? The new color computers are more limited than the older ones were. Most of the chips are soldered directly without sockets. Therefore upgrades that require removing a chip and plugging in a board or module will not be feasible. The only alternative is upgrades that use the expansion port. These will be more expensive. The cost of computers has gone down because no sockets were used causing the cost of hardware upgrades to increase.

Our new products, product review, and subscription drive is going well. I want to thank those who have assisted us in these areas.

In this issue we are including a cumulative index of all our issues. Some readers have written and asked when we covered certain topics. All back issues are available if you need them. We only have a small supply left of Vol. 1, No. 2.

WRITING PROGRAMS
(PART 7)

This is a series on writing basic programs. Each month we

give new material and give example programs on how to use the material. Last month we gave an address file program. In it we used arrays which allow information to be placed into rows and columns. An array is defined by the DIM statement. Also we have been looking at IF-THEN tests. These are decision making commands which cause the computer to branch to different locations depending upon the outcome of the test. Let's look at the IF-THEN test a little more because it is a very powerful programming tool.

IF-THEN TESTS

An IF-THEN test works like this. IF (this condition is true) THEN (do this basic command). A real advantage of this test is that a series of functions can be performed if they are all on one line number. Look at the following statements.

```
30 IF A=5 THEN X=75: J=53:
   P=25: SOUND 100,5
```

If A=5 then the following things happen.

1. X=75
2. J=53
3. P=25
4. SOUND 100,5

Now if A is not equal to 5 then none of the commands in statement 30 are executed and the program goes to the next statement.

IF-THEN ELSE

Several IF-THEN statements can be combined by using ELSE. Look at the following:

```
50 IF P=25 THEN A=31 ELSE IF
   P=50 THEN A=85 ELSE IF P=75
   THEN A=130
```

We can put these results in a table.

P	A
25	31
50	85
75	130

If P is any other value then A is not changed. Symbols used for IF tests are:

< Less than
> Greater than
= Equal
<= Less than or equal
>= Greater than or equal
<> or >< Not equal

BRANCHING

IF-THEN tests can also be used for branching. Consider the following:

```
30 IF X>50 THEN 500
```

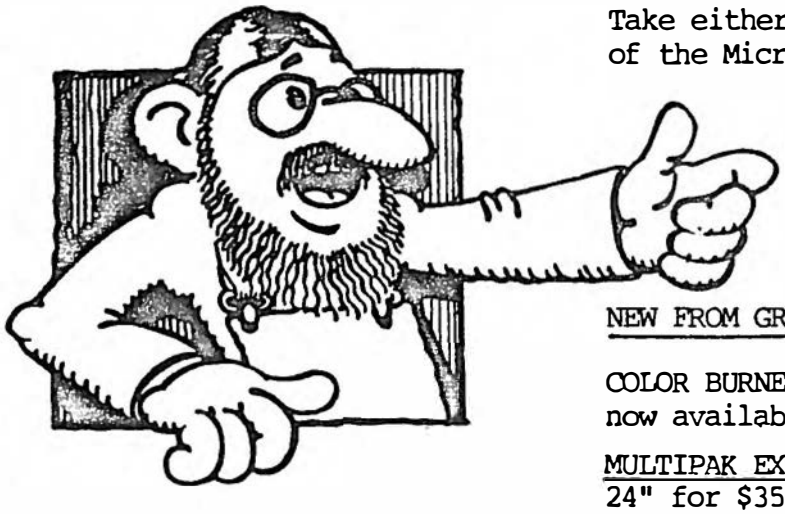
This means to go to statement 500 if X is greater than 50. If the test fails or X is not greater than 50 then the next statement is executed.

NEW MATERIAL

Let's look at some commands for operating on strings. Our objective will be to develop a technique for storing characters in memory. Now computers don't store characters, they only store numbers. So we have to store a number that represents a character. These numbers have been defined by the American Standard CODE for Information Interchange (ASCII). ASCII is pronounced as ASK-KEY with only one K. Each key on the keyboard is given an ASCII number. For example an "A" is 65 and a "0" is 48.

CHARACTERS to ASCII

It is necessary to convert keyboard characters to ASCII so the ASCII value can be stored in memory. Look at the following



SPECIALS for readers of DYNAMIC COLOR NEWS!

Take either a FREE 64K UPRGRADE with purchase of the Micro Language Lab Learning the 6809 course *OR* take 10% OFF Micro Language Lab!

Take a FREE COLOR QUAVER with the purchase of an assembled and tested CoCoPort input/output module!

NEW FROM GREEN MOUNTAIN MICRO!

COLOR BURNER with TEXTOOL ZIF SOCKET now available for only $10 extra!

MULTIPAK EXTENDER CABLES 18" for $30, 24" for $35, 36" for $40 (gold ends!)

Green Mountain Micro

MICRO LANGUAGE LAB: LEARNING THE 6809

Feeling at the mercy of a programmer somewhere? Mystified by a machine? My Micro Language Lab will give you the power to end those helpless feelings.

Here's why. The heart of *any* computer program *anywhere* is machine language. Every piece of software uses it — your favorite game, spreadsheet, word processor, data base manager, or recipe file. Every time you hit "Enter", it's working. All the languages and operating systems are created from it. Basic, Fortran, Pascal, Flex, OS-9, TRSDOS... they're 6809 machine language, the whole lot!

And you can learn the language of the 6809, the programming heart of your Color Computer, with my Micro Language Lab.

Not everyone can program. Writing a few lines in Basic now and then doesn't make you a programmer. But if you *can* program, then my Micro Language Lab will teach you — the right stuff, the right way.

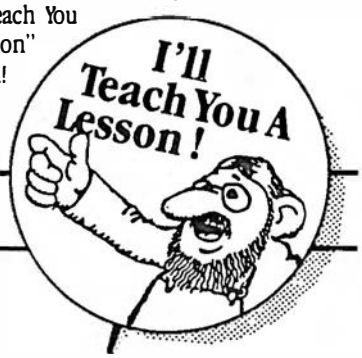
I'll teach you in 24 half-hour lessons on 12 audio cassettes, with a 220-page textbook, with data booklets, with 35 sample programs, and with a programming reference card. You'll spend 50 hours or more with my course, listening, watching, and working. And when you're done, you'll be programming your Color Computer in the 6809's language.

Stop ignoring those gnawing feelings of helplessness. The 6809 is where the power lies in your computer. Turn on that power with the Micro Language Lab.

- **Micro Language Lab,** $99.00 (plus $3.50 shipping and handling)

Requires 16K Extended Basic EDTASM +

Not sure? Write or call for a Table of Contents and sample pages — and your "I'll Teach You A Lesson" button!



ALSO AVAILABLE FROM GREEN MOUNTAIN MICRO

~~$2.50~~ $3.00 *(Add ~~$2.50~~ shipping and handling to your order)*

Lowerkit III

- Full-time upper and lowercase installs in 15 minutes.
- Normal and reverse video standard
- Fully compatible with all Alpha and Graphic modes

~~$79.95~~ assembled and tested $67.50
$49.95 complete kit of parts

Important! Specify Color Computer or Color Computer II

TV Buff II, clear image for video monitors, $19.95 (specify CoCo or CoCo2)

CoCoPort parallel interface (now includes plastic case), $54.95 / $44.95 kit

ColorPack ROM/RAM pack, $29.95 (specify configuration) / $19.95 kit / case $3.00 extra NEW DESIGN!

64K memory upgrade kit with full instructions, ~~$49.95~~ / with memory test on tape, ~~$54.95~~ $20.00 / $25.00

Color Quaver Software Music Synthesizer on tape (requires 32/64K) ~~$19.95~~ $14.95

Color Burner EPROM Programmer (2716/32/32A/64/64A/128, 68764/66) with software, $69.95 / $56.95 kit

User Group, Educational, Club and Dealer Discounts are available.

TRS-80 and TRSDOS are trademarks of Tandy Corporation. Flex is a trademark of TSC, Inc. OS-9 is a trademark of Microware Corporation.

Green Mountain Micro
Bathory Road, Box U
Roxbury, Vermont 05669
802 485-6112

Hours: 9am - 5pm EST, Monday - Friday
COD/VISA/MASTERCARD

program:

```
10 A$=INKEY$: IF A$="" THEN 10
20 A=ASC (A$)
```

Statement 10 has been used before to force the computer to wait for a keyboard entry. This is the string A$. When a key is pressed the computer goes to statement 20 and the numerical variable "A" is the ASCII value of the pressed key. Now that we have a number, it can be stored in memory.

ASCII to CHARACTERS

To know what character we stored in memory it is necessary to convert the numbers into a one character string. Assume that the numerical variable "A" represents the value of the character. The following will convert "A" into a character string A$:

```
50 A$=CHR$(A)
```

A$ can be printed on the screen or to an external printer.

CONVERTING STRINGS to VALUES

Before a character can be stored it must be converted to a numerical value. There is a basic command that will do this. It works as follows:

```
30 A= ASC (A$)
```

If A$ is the character then A is its ASCII value.

ASCII DEMO PROGRAM

The following program will allow you to observe the values of the keyboard keys. It will display the character and its ASCII value.

```
10 A$=INKEY$: IF A$="" THEN 10
20 A=ASC(A$):?A; A$: GO TO 10
```

Statement 10 forces the computer to wait until a key is pressed. When the key is pressed, this is A$. Statement 20 converts the string A$ to its ASCII value A, prints A and A$ and returns to statement 10 for the next character. Type in all of the numbers, punctuation, upper and lower case letters, plus the special characters and you will get a good idea of the ASCII values of the keys.

PEEKS & POKES

The POKE command is used to store values in memory and the PEEK command is used to recover the values from memory.

```
10 POKE M,V
20 X=PEEK(M)
```

Statement 10 is an example of how to use the POKE command. M is the memory location and V is the value from 0 to 255. Notice that a comma seperates the variables. Statement 20 is an example of recovering a value from memory. X will be the value and M is the memory. Notice that parenthesis are around M.

Next month we will show how to write a simple word processor program where we will store characters in memory and recover them. In this series we have been developing programming tools and will soon be able to write some very powerful programs with these tools.

FAST FOOD PROGRAM

You are ready to open your fast food restaurant and now have your computer program ready for the first customer. This program only uses concepts that we have covered in our program writing series. You can enter up to 50 items and select the number to represent the item in an array. The name of the item plus the cost is contained in

arrays. The program calculates the total cost plus 7% sales tax. You can modify this for your state. Comments are included to explain what each section does.

This program is mainly an application on using arrays, printing menus, and using logical tests. We started with the data and decided to enter one item per statement with the name and cost. Other items can be added and put in the order you would like for them to be displayed. Here come the customers so get ready.

FAST FOOD
PROGAM LISTING

```
10 ?"THIS IS A FAST FOOD
     PROGRAM"
20 ?"COPYRIGHT (c) 1985
30 ?"dYNAMIC eLECTRONICS iNC.
40 ?"THIS IS PGM 9-2-85
50 DIM N$(50), X(50),C(50),
   Y(50)
60 'READ DATA INTO THE ARRAYS
70 FOR K=1 TO 50
80 READ N$(K),C(K)
90 'EXIT IF EMPTY DATA
100 IF N$(K)="" THEN 130
110 PRINTC(K);N$(K)
120 NEXT K
130 N=K' SAVE THE NUMBER OF
     ITEMS
140 CLS:PRINT"THIS INITIALIZES
     THE ORDER
150 'THE ARRAY X(K) IS THE
     NUMBER OF THE ITEM IN THE
     ORDER
160 FOR K=1 TO N:X(K)=0: NEXT K
170 'SET UP FOR 20 ITEMS PER
     ORDER
180 FOR W=1 TO 20
190 'THE SUBROUTINE SELECTS THE
     ITEMS
200 GO SUB 540
210 IF X>N THEN 200
220 X(W)=X
230 'DISPLAY THE ITEM SELECTED
240 PRINT N$(X)
250 INPUT"HOW MANY";Y(W)
260 CLS
270 T=0:FOR Q=1 TO W
280 'CALCULATE THE TOTAL
290 T=T+C(X(Q))*Y(Q)
300 'PRINT THE TOTALS
310 PRINTY(Q);N$(X(Q));
    C(X(Q)) * Y(Q)
320 NEXT Q
330 PRINT"TOTAL ORDER="T
340 'FIGURE TAX AND ROUND IT
     OFF
350 TX=.07*T:TX=INT (100*TX)
    /100: PRINT"TAX="TX
360 'CALCULATE TOTAL WITH TAX
370 TT=T+TX
380 PRINT"TOTAL ="TT
390 PRINT"PRESS KEY TO CONTINUE
400 PRINT"PRESS N FOR NEW ORDER
410 PRINT"PRESS E TO END
420 Z$=INKEY$:IF Z$=""THEN 420
430 IFZ$="E" THEN INPUT"AMOUNT
    RECEIVED";A:GO TO 480
440 'START OVER FOR NEW ORDER
450 IF Z$="N" THEN 140
460 NEXT W
470 'FIGURE THE CHANGE
480 C=A-TT:PRINT"CHANGE DUE="C
490 PRINT"PRESS A KEY TO
    CONTINUE
500 'MAKE THE COMPUTER WAIT FOR
    YOU
510 P$=INKEY$:IF P$="" THEN 510
520 GO TO 140
530 'CLEAR THE ITEM COUNTER (V)
540 V=0:FOR J=1 TO N
550 V=V+1
560 PRINTJ;N$(J);C(J)
570 'BRANCH OUT IF WE DISPLAYED
    12 ITEMS
580 IF V=12 THEN 600
590 NEXT J
600 INPUT"ENTER NUMBER
    NUMBER";X
610 'IF X>0 THEN WE SELECTED AN
     ITEM
620 IF X>0 THEN RETURN
630 'RETURN IF OUT OF ITEMS J>N
640 IF J>N THEN RETURN
650 'RESET ITEM COUNTER AND
     CONTINUE
660 V=0:GO TO 590
1000 'ENTER DATA HERE
1010 'ENTER ITEM, PRICE IN DATA
      STATEMENTS
1020 DATA SMALL HAMBURGER,.59
1030 DATA LARGE HAMBURGER,.79
1040 DATA SMALL FRENCH
     FRIES,.59
1050 DATA LARGE FRENCH
     FRIES,.79
1060 DATA SMALL CHICKEN DINNER,
     1.79
```

```
1070 DATA LARGE CHICKEN DINNER,
     2.59
1080 DATA SMALL COKE,.69
1090 DATA MEDIUM COKE,.79
1100 DATA LARGE COKE,.89
1110 DATA SMALL PEPSI,.69
1120 DATA MEDIUM PEPSI,.79
1130 DATA LARGE PEPSI,.89
1140 DATA SMALL COFFEE,.49
1150 DATA LARGE COFFEE,.69
1160 DATA MILK,.59
1170 DATA ORANGE JUICE,.59
1180 DATA PECAN PIE,.79
1190 DATA LEMON PIE,.79
1200 DATA CHOCOLATE PIE,.79
1210 DATA SMALL PIZZA,3.50
1220 DATA MEDIUM PIZZA,4.50
1230 DATA LARGE PIZZA,5.75
1240 DATA VANILLA SHAKE,.89
1250 DATA STRAWBERRY SHAKE,.89
1260 DATA CHOCOLATE SHAKE,.89
1270 DATA ,,,
```

COMPUTER GRAPHICS
(Part 8)

Last month we looked at the line drawing command. We want to continue with this and develop a program that will present events or data in a graphic form. Bar graphs are very useful for statistical data. For example we could present a history of our electricity costs over the past 12 months. Or we could plot the value of stocks for each month over a 12 month or longer period. Of course there are many applications where it would be useful to present data in a graphic form. Bar graphs are very pleasing to look at because the value of a data element is represented by the height of the bar. We can look at the bars and observe quickly the changes that were made from one sample interval to the next.

LINE COMMAND

Graphic elements are defined as follows:

(X,Y) - Horizontal, Vertical
(0,0) - Upper left element
(255,0) - Upper right element
(0,191) - Lower left element
(255,191) - Lower right element

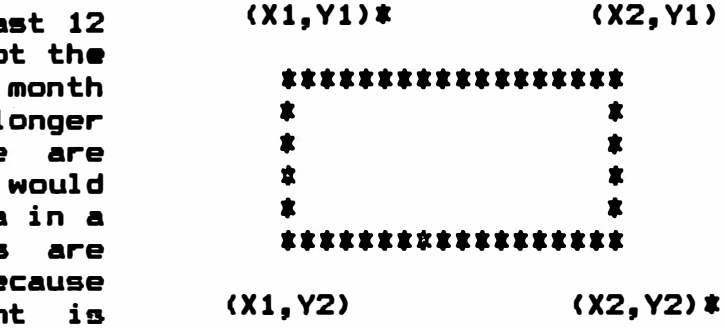
If we designate the two points that define a line as (X1,Y1) and (X2,Y2) then the line command is

```
LINE (X1,Y1)-(X2,Y2), PSET
```

To draw horizontal lines Y1 and Y2 have to be equal and to draw vertical lines, X1 and X2 have to be equal. Last month we gave instructions and a demonstration program forl using this command. This month we want to continue and show how to draw boxes and bar graphs.

DRAWING BOXES

By adding an extension to the line command we can draw boxes. The box will be a rectangle with the two points at opposite angles from each other. It takes 4 points to draw a rectangle as follows:



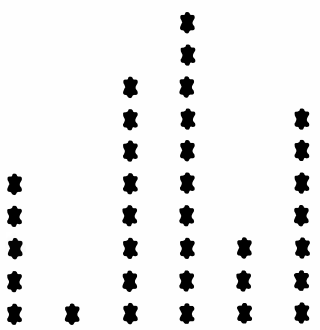
The rectangle or box is drawn through the two points with a * by them. The command for doing this is:

```
LINE (X1,Y1) - (X2,Y2), PSET, B
```

Notice the "B" at the end of the command. Think of this as representing "BOX". To fill in the box just add BF instead of B at the end of the command.

BAR GRAPHS

For those who are not familiar with graphs we want to give an example of a bar graph. These can easily be drawn by using a series of filled in boxes. Suppose we have data of 5,1,8,10,3, and 7 and want to present it in bar graph form. We will again use a "*" to represent a unit. The first data will have 5 stars, the second will have 1 star, etc. The final result will be similar to the following:



BAR GRAPH

Notice how easy it is to see how the data changed. We need to label the axes which we will do next month. We wrote a program to demonstrate how to draw boxes, filled boxes, and bars. This program is in the next section.

BAR GRAPH
DEMO PROGRAM

The following program allows boxes, filled boxes, and bar graphs to be drawn using high resolution graphics. This demonstrates the principles covered in the preceeding section. Comments are included to explain what each section does. Values are stored in memory from 500-525 so that they are not lost if you stop the program and then run it again.

```
 10 PCLEAR8
 20 CLS
 30 ?"BAR GRAPH DEMO PROGRAM
 40 ?"COPYRIGHT (c) 1985
 50 ?"dYNAMIC eLECTRONICS iNC.
 60 PRINT"PROGRAM 9-1-85
 70 ?"THIS PROGRAM DEMONSTRATES
    HOW TO DRAW LINES, BOXES,
    AND BARS.
 80 PRINT"USING EXTENDED COLOR
    GRAPHICS.
 90 PRINT"  - PRESS A KEY TO
    CONTINUE -":80 SUB 450
100 'SET UP ARRAYS
110 DIM X(5),X$(5),Y(15)
120 X$(1)=" X1":X$(2)="Y1";
    X$(3)=" X2":X$(4)=" Y2";
    X$(5)=" COLOR"
130 'DEFINE STRING ARRAYS
140 PCLS:FOR J=1 TO 5
150 'GET X(J) ARRAY VALUES
160 X(J)=PEEK(500+J)
170 NEXT J
180 CLS
190 'PRINT VALUES
200 ?"1" X$(1); X(1),"2"X$(2);
    X(2)
210 ?"3"X$(3);X(3),
    "4"X$(4);X(4)
220 PRINT"5"X$(5);X(5)
230 PRINT"6 DRAW BAR GRAPHS
240 PRINT"7 DRAW BOX
250 PRINT"8 DRAW FILLED IN BOX
260 'C IS THE COLOR
270 C=X(5)
280 PRINT
290 ?"ENTER NUMBER FOR OPTION
300 P$=INKEY$:IF P$="" THEN 300
310 N=VAL(P$)
320 'PRINT AGAIN IF N=0 OR IS
     GREATER THAN 9
330 IF N=0 THEN 140 ELSE IF N>9
    THEN 140
340 'BRANCH TO APPROPRIATE
     SECTIONS FOR N
350 IF N=6 THEN 490
360 IF N=7 THEN 410 ELSE IF N=8
    THEN 470
370 'N IS FROM 1 TO 5: ENTER
     NEW VALUE
380 ?"ENTER NEW VALUE FOR"X$(N)
390 INPUT X: POKE 500+N,X
400 GO TO 140
410 COLOR C,1:PMODE 3,1:SCREEN
    1,0:LINE (X(1),X(2)) -
    (X(3),X(4)), PSET, B
420 'WAIT FOR KEY TO BE PRESSED
430 GO SUB 450
440 GO TO 140
450 C$=INKEY$:IF C$=""THEN 450
460 RETURN
470 COLOR C,1:PMODE 3,1:SCREEN
    1,0:LINE (X(1),X(2)) -
    (X(3),X(4)),PSET,BF
480 GO SUB 450: GO TO 140
490 PRINT"THIS DRAWS BAR GRAPHS
500 'CLEAR GRAPHICS AND SCREEN
510 PCLS:CLS
520 'DISPLAY BAR GRAPH VALUES
    FROM MEMORY
530 FOR K=1 TO 12
540 Y(K)=PEEK(510+K)
550 IF Y(K)>191 THEN PRINT
    "VALUE GREATER THAN 191
560 'TAKE CARE OF THE A,B,& C
```

```
    OPTIONS
570 IF K=10 THEN PRINT"A";:
    GO TO 620
580 IF K=11 THEN PRINT"B";:
    GO TO 620
590 IF K=12 THEN PRINT"C";:
    GO TO 620
600 'LABEL AND PRINT THE VALUES
610 PRINTK;
620 PRINTY(K)
630 NEXT K
640 PRINT"ENTER CHARACTER TO
    CHANGE OR PRESS ENTER KEY
    TO DRAW BAR GRAPH.
650 GO SUB 450:N=VAL(C$)
660 'DEFINE N FOR C$>9
670 IF C$="A" THEN N=10 ELSE IF
    C$="B" THEN N=11 ELSE IF
    C$="C" THEN N=12
680 'BRANCH IF VALUE TOO LARGE
     OR TOO SMALL
690 IF N>12 THEN 490 ELSE IF
    N=0 THEN 730
700 ?"ENTER NEW VALUE FOR "C$
710 INPUT X:POKE 510 +N,X:
    GO TO 490
720 'ENTER GRAPHICS MODE
730 PMODE 3,1:SCREEN 1,0
740 COLOR C,1
750 'DRAW BAR GRAPH
760 FOR Q=1 TO 12
770 'DEFINE Y2 SO GRAPH STARTS
     FROM THE BOTTOM UP
780 Y2=191-Y(Q)
790 X1=20*Q:X2=X1+4
800 LINE (X1,191) - (X2,Y2),
    PSET,BF
810 NEXT Q
820 GO SUB 450:GO TO 140
```

LARGE MEMORY PROGRAMS
(Part 8)

How can you use the extra memory in your computer? The hardware in color computers only allow 32K bytes to normally be used for random access memory (RAM). The synchronous address multiplexer (SAM) chip or MC6883 is designed to allow two modes of operation or configurations for 64K RAMS. We want to review these modes.

The two memory configurations are called memory type 0 and memory type 1. Memory type 1 configures the computer for the all RAM mode. To use this all of the read only memories (ROMS) have to be copied into RAM which means you must have a 64K computer. In this mode you can reconfigure your operating system which is what DOS programs do. Some word processors, spread sheets, and games also use this configuration because it gives about 8K more of usable memory. We showed how to do this and we developed a program for this purpose which was printed in an earlier edition of this newsletter. This program is included in our DCN-1 package.

Memory type 0 allows the computer to address a total of 96K bytes. The lower 32K bytes is RAM and the upper 32K bytes is ROM. A powerful feature of this configuration is that the RAM is grouped into two 32K bytes which can be selected by memory pokes. In previous editorials in this series we showed how to use the other 32K memory bank. In fact we developed a program to do this and have included it in our DCN-1 package. The problem with using this second bank is initializing it for basic. We showed how to initialize the second bank by copying each byte from the first bank into the second bank.

RAM DISK DEVELOPMENT

Last month we started looking at what would be required to just use the second bank for storing programs. For each program we will need to know its beginning (2), its ending (2), its execution address (2), its name (8), its beginning location in RAM (2), and its ending location in RAM (2) plus the type of program (1). The numbers in parenthesis represent the bytes required. We decided that we would allocate 20 bytes for each program.

How many programs can we store in a 32K bank? If we have large programs we can not store very many but maybe we should allow for some short machine language and basic programs. So let's allow for 20 programs. This means that we will need about 400 bytes for our program directory.

BASIC VECTORS

Machine language programs do not present any problems. We just load them into the designated area and run them. Basic programs are not that easy. Preceeding each statement is a zero. The first two bytes contain a vector that points to the memory location of the next statement. If the memory location the program is being loaded into is different from the memory it occupied when it was saved, then the vectors will be for the old memory location. To give an example suppose you had reserved 8 graphic pages by executing "PCLEAR 8". This program resides in the upper part of the 32K RAM. Now when this program is saved to the second bank the vectors are for the PCLEAR 8 condition. Suppose you enter PCLEAR 1 and transfer the program from the second bank into the memory for a PCLEAR 1 condition. The vectors will need to be changed. If this is not clear do a PCLEAR1 and PEEK (25). Notice the value. Then do a PCLEAR 8 and PEEK (25). Notice that this value is higher. The vector in 25 and 26 point to the start of a basic program. The decimal value of the start of the program is calculated by

V = 256 * PEEK (25)+ PEEK (26).

```
+ ++ + ++ + ++ ++ ++ + ++ + ++ +
+                               +
+         RENEWAL TIME?         +
+                               +
+ The date beside your name on +
+ the address label indcates   +
+ the last issue you will re-  +
+ ceive. Send in your renewal  +
+ if you want to continue re-  +
+ ceiving technical informa-   +
+ tion on Color Computers.     +
+ This is the last issue for   +
+ those with 9/85.             +
+                               +
+ ++ + ++ + ++ ++ ++ + ++ + ++ +
```

CORRECTING BASIC VECTORS

The first two bytes of a basic statement are a vector that point to the memory location of the next basic statement. As stated earlier a zero preceeds each basic statement. To correct the vectors we can look through the memory until a zero is found. The next location is the beginning of the next statement. The next statement vector for the preceeding statement should point to this location.

As an example suppose we have a program with a vector in 25 pointing to 3585. We look at memory starting at 3584 and notice the following decimal values.

Memory	Value	
3584	0	PRECEEDS STMT
3585	14	POINTS to 3592
3586	8	
3587	0	STATEMENT 1
3588	1	
3589	58	
3590	131	
3591	0	PRECEEDS STMT
3592	14	POINTS to 3627
3593	43	
3594	0	STATEMENT 2
3595	2	
3596	58	
3597	131	

For the above location 3584 contains a zero which preceeds the first statement. 3585 & 3586 contains the vector 14,8 which points to 3592 which is the beginning of the next statement. The statement number vector is in the third and fourth locations. Notice the zero in 3591 indicating the end of the previous statement.

We have written a basic progam that will correct statement vectors. This program is slow but demonstrates what is happening. For our final version, we will want a machine language program.

VECTOR CORRECTOR PROGRAM

```
10 PRINT"BASIC PROGRAM VECTOR
   CORRECTOR
20 PRINT"PGM 9-3-85
```

```
**********************************
*                                *
*          BACK ISSUES           *
*                                *
*  Back issues of DCN are        *
*  available for $1.95 each,     *
*  3 for $5 or 12 for $15 pp.    *
*                                *
*  Foreigners other than Can-    *
*    ada add $2 for Air Mail     *
*            postage.            *
*                                *
**********************************
```

```
30 PRINT"COPYRIGHT (c) 1985
40 PRINT"dYNAMIC eLECRONIC iNC
50 PRINT
60 ?"THIS CORRECTS STATEMENT
   VECTORS FOR THE MEMORY
   LOCATION IN WHICH THE
   PROGRAM RESIDES.
70 ?"ENTER MOST SIGNIFICANT
   BYTE FOR PROGRAM LOCATION.
80 INPUT MS
90 'SAVE PROGRAM LOCATION IN
    500
100 POKE 500,MS
110 LS=256*MS+1'LS STANDS FOR
    LAST STATEMENT
120 'SKIP 4 BYTES AND START
     LOOKING AT MEMORY
130 M=LS+3
140 M=M+1:A=PEEK(M)
150 PRINT"M="M;"A="A:IF A>0
    THEN 140
160 'A=0: SAVE THE NEXT
     STATEMENT LOCATION (NS)
170 M=M+1:NS=M:M1=INT(NS/256):
    L1=NS-256*M1
180 SOUND 100,5
190 PRINT"NEXT STATEMENT="NS
200 PRINT"THIS VALUE IS BEING
    PUT INTO"LS
210 'POKE VECTOR VALUES INTO
     FIRST TWO BYTES OF
     PRECEEDING STATEMENT
220 POKE LS,M1: POKE LS+1,L1
230 'LOOK FOR END OF PROGRAM
     INDICATED BY NS=0
240 IF NS=0 THEN 280
250 'NS IS NOT ZERO
260 'LET LS=MEMORY LOCATION
270 LS=M: GO TO 130
280 PRINT"WE HAVE FINISHED. TO
    RUN THE
290 PRINT"PROGRAM MOVE THE
    BEGINNING OF
300 PRINT"PROGRAM VALUE
    ORIGINALLY SAVED
310 PRINT"IN 500 TO 25 AND YOU
    CAN RUN
320 PRINT"THE CORRECTED PROGRAM
```

PRODUCT REVIEWS

This section is open to all producers and dealers of color computer products. We will review you product free of charge and write an editorial on the product. We do not use a rating system but will explain what the product does, and what can be expected from it. Any comments about the review from the firm submitting the product will be printed in a later issue.

* * * * * * * * * * * * * * * * * *

FULL CHARACTER SET for COCO or COCO 2

One of the undersireable features of Radio Shack Color Computers is the lack of lower case characters. Lower case characters are represented by reversing the characters. This is a limitation of the MC6847 video display generator (VDG) which is manufactured by Motorola. Motorola has developed a new VDG chip that has true lower case characters but it is not a direct replacement for the MC6847.

CoCo Devices has designed a plug in board for color computers with sockets for the MC6847 chip. This board contains several electronic chips plus an EPROM with a new character generator. Several versions are available for the different color computer series. We were sent one for our 285 computer which we used for this review.

To install the board turn the

power off, remove the covers and locate the MC6847 chip. Remove this chip and plug the board into the socket. Plug the MC6847 into the socket on the board. You can now turn on the computer and type "SHIFT 0" to enable the lower case characters. Type in a few characters and you can determine if the board is working properly. Then turn off the computer and replace the covers.

Let's look at the features of the character set board.

1. It displays all 96 standard ASCII characters.
2. Upper and lower case are displayed simultaneously.
3. True descenders on lower case characters.
4. Braces and vertical bar characters
5. Slashed zero to distinguish from letter "O".

Additional hardware features allow wiring the board for normal or inverted video, and returning the display to the original character set. A switch or switches can be added if it is desireable to select these features.

OPERATION

The character generator board does not alter the operation of the computer. Once it is installed the computer can be operated as if it didn't exist except lower case characters are displayed as lower case characters and not reversed capital characters. If you are used to the reversed characters, than the display will seem a little strange at first. However it relieves strain by having all of the characters displayed in the same format.

The design of word processors could be greatly simplified with the character generator board. One of the functions of most word processors is to provide a character generator set using the graphics modes of the computer. With the character generator board installed, this would not be required simplifying the design of word processors and freeing extra memory for text.

SUMMARY

We found the character generator set very enjoyable to use. The installation was easy and the board is constructed of excellent material. This is an economical way to add lower case characters to computers with sockets for the MC6847 chip.

CoCo Devices, Box 677, Seabrook, TX. 77586 (713) 474-3232, $38 + $2 S/H.

- - DCN Staff - -

TELEPATCH
TELEWRITER ENHANCER

Telepatch is an enhancer for the Telewriter Word Processor. It adds some very useful features that are not available on the Telewriter disk program. Telepatch comes on a disk with 12 pages of instruction. It incorporates the improvements into a new program which can be saved onto a disk.

The telepatch disk should be copied onto a new disk so as not to destroy the original. A disk with an original telewriter program is required as well as a formatted disk for the enhanced program.

After the backup disk is made, several disk swaps are required as the new enhanced program is merged with the original telewriter program. We experienced no problems with this procedure. The new disk contained the following programs:

T/BAS
*/***

TW64/NEW
TSPOOL/BAS

To run the new program just type "RUN T". Now let's look at the features of Telepatch.

1. True block move. Telewriter only had a block copy feature.
2. Visible carriage returns. This is enabled by using "CLEAR I".
3. Key Beep. "CLEAR L" allows a beep to be heard in the speaker when each key is pressed.
4. Key Repeat. All keys will repeat if they are held down for more than about a half second.
5. Justify. Leaves justify on after files are printed.
6. Reset. Returns to the editor when the reset button is pressed.
7. Disk I/O. This is stored in RAM and does not have to be loaded from disk.
8. Lowercase. For the editor lowercase is "ON" and for disk menu lowercase is "OFF".
9. Eps/ok 1f fonts. Corrects error in setting font buffer.
10. Fast cursor. Speeds up cursor movement when using SHIFT and an ARROW key.
11. Overstrike mode. Allows characters to be rewritten over existing characters.

MEMORY POKES

The new boot program "T/BAS" is modifiable to allow changing the initial values in the menus. The instructions tell which line numbers are used for the various parameters. For example you can set your printer's baud rate by modifying statement 1190.

DISK DRIVER

The ASCII and BIN disk drivers have been combined into one disk driver which resides in RAM. You can switch from one to the other by pressing "I". The ASCII driver uses the extension "TXT" for I/O and the BINARY driver uses the extension "BIN". Other extensions can be used by typing them in. A nice feature of the disk driver menu is that text can be saved using the last filename without having to retype it. This works with READ, SAVE, APPEND, and PARTIAL SAVES. To use this just press "ENTER" when normally you would type in a file name. You will then be asked for y/n. Enter "Y" and the last filename will be used.

OPERATION

After formatting a disk as previously discussed, the program is run by typing "RUN T". This loads everything including the disk I/O program which resides in RAM. The first menu appears on the screen and you can select the print menu, disk I/O menu or the editor. This is similar to the Telewriter menu. Switching between the menus is quick and easy since no disk I/O programs have to be loaded as was the case with the original program.

In the editor there are two things that are different. The first is that lower case characters are turned on. This saves time because generally you do not want all capitals in your text and would previously have to type "SHIFT O". The second thing is that the auto repeat key is enabled. This is a powerful feature because it lets you quickly move to different parts of your text with the arrow keys and allows rapid insertion of the same character.

Visible markers are available to mark the end of each line. They are activated by entering "CLEAR I" and deleted by the same command. This helps spot unnecessary spaces at the end of

the line.

You can hear a click in the speaker everytime you press a key. This is enabled and disabled by entering "CLEAR L". With the clicks the operator has a feedback from the computer as each character is entered. This lets him know that the computer has received the character.

A very useful feature of the editor is the block move. To move a block mark the beginning and ending of the block with "CLEAR B" and "CLEAR E". Then move the cursor to the location where the block is to be finally located. Then enter "CLEAR T" and the block is moved. With the original program it was necessary to go back and delete the block from its original location. This step is not required with the enhanced program.

Another addition to the editor is the overstrike mode. With this you can write new characters over previous characters. This feature is enabled and disabled by entering "CLEAR O".

SUMMARY

The RESET feature did not work for us. A hard reset occurred when we pressed the reset button. We did not find this to be a disadvantage because the text can be quickly saved and reloaded. The tape I/O features were deleted so if you want the tape feature you will have to use the original program.

As a bonus the program "TSPOOL" allows text to be printed to the disk instead of a printer. It can then be reloaded and will appear in the editor the same as it would be printed with the right margins justified. We used TSPOOL in writing this newsletter so we could see how editorials would appear when printed.

The Telewriter Enhancer makes a good word processor much better. The addition of auto repeat, true block move, key clicks, and visible markers greatly improve the editor. The enhancer speeds up the operation when scrolling through text, changing menus, and reading in programs.

Spectrum Projects, P. O. Box 21272, 93-15 86th Dr., Woodhaven, NY 11421, $19.95 + $3 S/H.

- - DCN STAFF - -

```
* * * * * * * ** * * * * * * * *
*                               *
* DCN PROGRAMS on Tape or DISK  *
*                               *
* A collection of the programs  *
* from May, June, & July 1985   *
* DCN. The collection includes  *
*                               *
* 1. 64K All RAM Program        *
* 2. 2-Bank address file Pgm.   *
* 3. Alarm Clock Program        *
* 4. Loan Interest Program      *
* 5. Character Generator pgm.   *
* 6. Bank Switching Program     *
*    (Allows full use of other  *
*     32K bank for 64K comp.)   *
*                               *
*       Order DCN-1             *
*                               *
*   Tape $9.95, Disk $11.95     *
*  Add $2 shipping, Foreign $3  *
*                               *
* * * * * * * ** * * * * * * * *
```

NEW PRODUCTS

This section is available free for producers and dealers of color computer products. These products have not been reviewed by us but are included for our reader's information. Send a description of new products to:

New Products
Dynamic Electronics Inc.
P. O. Box 896
Hartselle, AL 35640

****

MAGIC LESSONS

Welcome to the wonderful world of magic! These new programs were written for those who wish to look like a professional without the years of practice.

Each program comes with three tricks. They are supplied with all the necessary props along with HI-RES text and graphic screens for ease of learning. Lessons 1-3 teach card tricks and lessons 4-6 teach coin tricks. 32K extended basic is required. Price $19.95 disk. For more information contact Merlin's Software, 11515 Casey Rd., Tampa, FL 33624, (813) 961-0135.

THUNDERRAM OS-9 RAMDISK DRIVER

The Os-9 Ramdisk Driver emulates a single-sided, 40 track floppy disk. Anything that you can do with a floppy disk, you can do with the Ramdisk. Instead of using "/D0", "/D1", etc. use the Ramdisk's name,"/R0".

The Ramdisk is compatible with all known utility commands and applications for OS-9. If you are using 40 track, single-sided floppies, you can even BACKUP between the floppies and the Ramdisk.

The big advantage to using a Ramdisk is SPEED! The total time to read/write a sector on the Ramdisk is less than 5/1,000 of a second!! This makes floppies look even slower than they actually are. Even if you own one of the Hard Disks available for CoCo, your Ramdisk will run rings around it. $24.95.

For more information contact Spectrum Projects, P. O. Box 21272, 93-15 86th Drive, Woodhaven, NY 11421.

CUMULATIVE INDEX

For

Dynamic Color News

February 1984 (Vol. 1 No. 1) Basic Prog., The Immediate Mode, Vectors; Multiprogram Manager; ML Prog., Microprocessor, Calling ML Programs from Basic; Q & A.

March 1984 (Vol. 1 No. 2) Basic prog., Variables and Arrays; Memory Expansion, 32K, 64K, 128K, 256K; ML, 2's Complement Numbers, Indexed Addressing; Oper. Hints; Q & A.

April 1984 (Vol. 1 No. 3) ASCII Part 1; Basic Prog., Arrays, Data and Read Statements; ML Prog., Data Relocation, Branching and Conditional Test, Compare Instr, Branch Instr; Oper Hints; Q & A.

May 1984 (Vol. 1 No. 4) ASCII Part 2; ML Prog., Interrupts, Enabling the Interrupts; Installing an Interrupt Switch; 128K Memory Expanders, Info. Editorial; 64K Memory Solution, Info Editorial; Utility Prog.; Oper. Hints; Q & A.

June/July 1984 (Vol.1 No. 5) Powerful Remarks-Part 1, WP, Statement Number Subroutine, Using Subroutines, WP Prog.; ASCII part 3, Interfacing ASCII Devices, Connecting two Color Computers Together; Uninterrupted Power Sources, Power Inverters, Motor Generators, Battery Backup Power; ML Prog, Interrupt Examples; Oper. Hints.

August 1984 (Vol. 1 No. 6) Powerful Remarks - Part 2, Data in Remarks; WP, Modifying WP, Writing Basic Programs, Composing Files, Writing Text; ML Prog., Prog. Development; Letters.

September 1984 (Vol.1 No. 7) ML Prog., Writing Position Independent Codes, Scratch Pad Memory, ML Prog. with fixed SP memories; Powerful Remarks - Part 3, Relocating the ML Subroutine, Calling the Subroutines from Basic; Q & A.

October 1984 (Vol. 1 No. 8) Basic Prog., Data Handling Techniques; ML Prog., Memory Searching, ML Subroutine, The Basic Program; 96KX Notes; Operating Hints.

November 1984 (Vol. 1 No. 9) Computer Generated Sound- Part 1; Basic Prog., Sorting Info,; Oper. Hints; Q & A.

December 1984 (Vol. 1 No. 10) Random Numbers, Random Sounds, Shuffling Cards; Computer Generated Sound - Part 2, CoCo Sound Waves; Sound Learning Prog., ML Subroutine; Sound Demonstration Prog., ML Subroutine, High Frequency Sound; Sound Prog. Listing; Oper. Hints.

January 1985 (Vol. 1 No. 11) Random Numbers - Part 2; Card Shuffling, Prog. Listing; Shuffling Program; Computer Sounds - Part 3, Sine Wave, Square Wave, Triangle Wave, ML Subroutine; Sound Program Discussion; Oper. Hints; Q & A.

February 1985 (Vol. 2 No. 1) Large Memory Programs - Part 1; Data Handling; Program Division; Seperate Data Files; The SAM Chip; SAM Pokes, Computer Graphics - Part 1; VDG Chip; How Video Works; Television or a Monitor?, Video Reverser - Hardware Project; Wiring Instructions.

March 1985 (Vol. 2 No. 2) Large Memory Programs - Part 2; Loading Subroutines into Both Banks, Bank Switching Subroutines, Bank Copy Subroutine, 96KX, Compatibility; Computer Graphics - Part 2, The Normal Display & Graphic Pages, Forming

Characters, Graphics Memory, Cursor Move Program; Cursor Demonstration Program Listing; Writing Programs - Part 1, Program Structure, NEW Command, Remarks, LIST & LLIST.

April 1985 (Vol. 2 No. 3) Large Memory Programs - Part 3; Transferring Variables Between Banks, 2 Bank Programs, 2 Bank Demonstration Program; Computer Graphics - Part 3, Semigraphics 4, Semigraphics 4 Demo Program; Writing Programs - Part 2; Variables; Numerical Variables; String Variables; Entering Variables; The Print Command; Print @ or ? @, Printing to a Printer, String Printing, Print Demo Program; New Products; Loan Interest Program.

May 1985 (Vol. 2 No. 4) Writing Programs - Part 3, Using Variables, INPUT Command; Gas Mileage Program; Large Memory Programs - Part 4, Vector Memory Locations, B0-B1 Block Move, B1-B0 Block Move; Data Move Program; Computer Graphics - Part 4, PCLEAR X, PMODE M,P, SCREEN T,S, GRAPHICS PAGES; New Products; Q & A; Operating Hint.

June 1985 (Vol. 2 No. 5) Writing Programs - Part 4, Variables, Entering Variables, Disadvantage of INPUT, Arrays, DIM Statement, FOR - NEXT Loops; Grade Book Program; Computer Graphics - Part 5, Character Generation, Data Approach; Graphics Demo Program (Character Generator); New Products

July 1985 (Vol. 2 No. 6) Large Memory Programs - Part 6, Stacking Programs, Switching Programs, 64K All RAM; 64K RAM Program; Writing Programs - Part 5, For - Next Loops, More on Arrays, Read & Data , Branching; Alarm Clock Program; Computer Graphics - Part 6, Generating Large Characters; Character Generator Program; COCO Heat Problem - Editorial by Dennis Kistz; Product Reviews, Spectrum DOS 1.0 - Spectrum Projects.

August 1985 (Vol. 2 No. 7) Writing Programs - Part 6 ; Arrays, Subroutines, Print Subroutine, Empty String; Address File Program; Large Memory Programs - Part 7, RAM Disk Development; Study Program; Computer Graphics - Part 7, The Line Command; Line Demo Program; Product Reviews, Thunder RAM - Spectrum Projects; New Products

September 1985 (Vol.2 No. 8) Writing Programs - Part 7; IF - THEN Tests, Branching, Characters to ASCII, ASCII to Char., Converting Strings to Values, ASCII Demo Pgm, PEEKS & POKES; Fast Food Program; Computer Graphics - Part 8, Line Command, Drawing Boxes, Bar Graphs; Bar Graph Demo Pgm; Large Memory Programs - Part 8, Ram Disk Development, Basic Vectors, Correcting Basic Vectors; Vector Corrector Program; Product Reviews; Full Character Set - CoCo Devices, Telepatch - Spectrum Projects; New Products.

* * * * * * * * * * * * * * * * * *

NEXT MONTH

Have you ever wished you had an extra port? Next month we will have a hardware project and show how you can add a second port to your computer. Also we will show you how to PCLEAR -1 and have over 32K of memory available when you ?MEM. We will continue with our series of editorials plus have more product reviews.

* * * * * * * * * * * * * * * * * *

CLASSIFIED ADS

1. 10 cents a word, $3 minimum.
2. Name, Address, & Telephone listed free.
3. Send payment with ad.
4. Closing date 1st of the preceeding month. Ex. Nov ad closing is Oct. 1.

* * * * * * * * * * * * * * * * *

DISPLAY ADS

Closing 1st of preceeding month.

Pages	1 time	2 times	3 times
1	$25	$22	$20
1/2	18	15	13
1/4	10	8	7

OPERATING HINTS

Print Disk Directory: Just enter "POKE 111, 254: DIR" and your disk directory will be printed.

Disable ROM Port: First poke 65314, 54 and a cartridge will not start when plugged into the ROM port. To run the program in the cartridge EXEC 49152. To restore the expansion port POKE 65315,52. Note plugging cartridges in with the power on can damage your computer.

Clear 0 pages: You can clear 0 pages by "POKE 25,6: NEW": This gives about 31K of memory but will not work for disk drives. You can increase your usable memory for disk drives by "POKE 25, 14: POKE 14 * 256,0: NEW".

Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I Want to receive instruction on programming, Computer Theory, Operating Techniques, Computer Expansion plus the Individual Reply to my Computer problems for a special of $10 each. Also I understand that there will be no charge for letters printed with answers in the Newsletter. Cost $15 USA & Canada, $30 foreign.

Name _____ Mail payment to
Address _____ Dynamic Electronics Inc
City _____ P. O. Box 896
State & Zip _____ Hartselle, AL 35640
Enclosed is a check ___
charge to VISA ___ MC ___ Number _____Exp._____

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