ENGINEERING NOTES
on
Radio Shack Color Computers

JUNE 1985
Vol. 2 No. 5

$1.95

DYNAMIC COLOR NEWS

* CHARACTER GENERATOR PROGRAM
* GRADE BOOK PROGRAM
* ADDRESS FILE PROGRAM

* LARGE MEMORY PROGRAMS (Part 5)
* WRITING PROGRAMS (Part 4)
* BASIC PROGRAMMING
* COMPUTER GRAPHICS (Part 5)
* QUESTIONS & ANSWERS
* OPERATING HINTS

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 *
* *
* June 1985 *
* *
* Editor and Publisher *
* Bill Chapple *
* *
* Secretary *
* Deanne Hill *
* *
****

CONTENTS

EDITOR'S COMMENTS

The hot Summer has arrived here in Alabama. The city swimming pool has not opened because of some renovations. I have noticed kids playing in small pools and people heading to the beaches. The season has changed here and so are computers. They are constantly going through changes and now with 128K, 256K, and larger memories available the possibilities for their use are greater than ever. We can make them play games, talk, generate music, control devices, do arithmetic calculations, process text, plus much more. The applications are numerous.

I have noticed that computers are not used very much in my town for unconventional applications. By unconventional applications I mean uses that do not involve calculations. I haven't gone into a store and seen a computer with details about the merchandise for sale on a screen. We know how to write messages on the screen and this seems to me like it would be very useful in a store for displaying messages to the customers. In this issue we show how to write messages on the screen by generating a character generator set with graphics. With one of the super large TV screens a lot of people could see the messages. This could be thought of as a programmabl sign. There are many fixed signs on which merchants change the letters for different messages. I don't know what these signs cost and guess that the cost varies depending on the size and quality of the sign. Writing messages on one of these large televisions is a very easy task for the color computer. It would not be hard to display the same message on several televisions at different locations within a store. To do this a radio frequency (rf) amplifier could supply a large enough signal on channel 3 or 4 for several television sets.

Most independent merchants ignore computers. I recently bought some building materials from a local lumber company. They had their prices listed in a notebook. They were handwritten and when the price changed they erased the price or wrote another page. There were several of these books and I can imagine what a job it is to keep these books current. Another lumber company I visited was completely computerized. The computer did all of the record keeping, kept track of stock, and gave the prices of each item. It also printed the invoice. This was obviously an advantage and I am sure that the computer's cost has been paid for by the savings in labor.

This month we are continuing our series on writing programs and computer graphics. For those of you who have not seen our previous issues, we have covered many topics and will try to publish an index of articles next month. Also we are considering putting some of these programs on tape.

Also we are continuing our series on writing programs. This is the same style of teaching we use in our Computer classes. Study each program segment that we cover and make sure you understand what we did. Then when we cover additional material you will be proficient in using the material covered.

PRODUCT REVIEWS

We have asked manufacturers and distributors of Color Computer products for products to review. We have received some and will start a review section next month. In this section we want to tell you what a product

will do and what you can expect from it. One of our products was reviewed by a magazines and got an unfavorable review because the reviewer did not like the way we did a particular routine. Personal opinions should not be reflected in a product review. So our reviews will explain what the product is advertised to do, and what you can expect it to do for you. Remember our product reviews will start next month and if you are a manufacturer or dealer send us you product for review. This service is free. Send products for review to:

DYNAMIC ELECTRONICS INC.
1001 Hwy 36 East
P. O. Box 896
Hartselle, AL 35640

WRITING PROGRAMS
(PART 4)

In preceeding editorials in this series we have showed how to use print commands and introduced variables. Perhaps these can be designated as the fundamental tools for programming. If you have results and don't know what they are then they are not of much use. So it is very nice to print results on the screen or an external printer. Also the results should be labeled so you will know which result is being printed. Supoose you have results printed on the screen as follows:

```
25              129.87
256             974
```

The above numbers may mean something to you if you just wrote the program. Would you know what they mean six months from now when you again run the program? It would be much more useful if results could be displayed as follows:

```
The number of players is 25
Uniforms cost $ 129.87
```

Each variable that is printed has a word phrase to describe it. If X represents the number of players and C represents the cost of the uniforms then we can write basic statements to print the results as follows:

```
10 ?"The number of players is"
     X
20 ?"Uniforms cost $" C
```

Notice that the word phrase is printed exactly as it appears between the parentheses. The symbol representing the variable is placed next to the word phrase. Basic prints the word phrase and then the value of the variable represented by the symbol.

VARIABLES

This month we want to concentrate on variables. Remember that there are two kinds of variables which are numeric and string. We can perform mathematical operations between numerical variables like we did last month with our gas mileage demonstration program. For string variables the only operation we can use is +. If we have two strings A$ and B$ defined as A$= "JAMES" and B$= "WATSON". Then we can do the following:

```
10 ?A$
20 ?B$
30 ?A$ + B$
```

Remember the ? means PRINT. When the program is run the following will be displayed.

```
JAMES
WATSON
JAMESWATSON
```

Statement 30 means to print the string A$ and then print the

string B$. Notice that there is no space between the strings. You have to add a space if you want one. To add a space we can modify statement 30 as follows:

```
30 ?A$ + " " + B$
```

When statement 30 is run the result will be

```
JAMES WATSON
```

ENTERING VARIABLES

How can variables be entered into the computer? There are several ways that this can be done. We can enter variables using the "INPUT" command. When this command is encountered the computer waits until the variable is entered from the keyboard. This is advantageous for some applications but will be awkward for others. You will want to use the input command for applications requiring a fresh supply of data. This is very nice for a checkbook program where you enter the value of each check. Also when you are running a program with many options you can select the one you want to run with an "INPUT" command. Of course there are many more applications where this will be desireable.

DISADVANTAGE of INPUT COMMAND

Suppose you have the same initial value for variables whenever a certain program is run. You would not want to enter these values everytime you run the program. So it would be nice if these variables could be contained within the program. There are two general methods of doing this.

The first method is to define the variables within program statements. We can write statements similar to the following:

```
10 A=25.36: B$="John Smith"
20 B=975.99: V$="CHECK BOOK"
```

We can put several commands within one statement number by seperating them with the ":". Notice that this method requires a lot of typing. Each symbol for the variable has to be typed followed by "=" marks. If it is a string variable then quotation marks have to surround the string.

The second method of defining variable is to use "READ" and "DATA" statements. We will be using both of these methods but first we want to discuss arrays.

ARRAYS

Arrays allow information to be put into rows and columns. Suppose you are a school teacher and have a grade book containing the names of the students and their grades for each test. The names of the students would be in the first column and the grades in the other columns. The first student would be in the first row, the second student the second row, etc. Instead of giving the students names, we will use numbers. If we have 30 students then they will be numbered from 1 to 30.

DIMENSION STATEMENT

The DIM or DIMENSION statement is used to designate arrays. We can have single or double arrays. For our class room example let's write a DIMENSION statement.

```
10 DIM N$(10),G(3,10)
```

In statement 10 we reserved 10 strings for the names of the students. Also we allowed an array to exist with 3 grades per student and 10 students. Our array will appear in the following format when printed.

```
N$(J)  G(1,J) G(2,J) G(3,J)
```

J will be the number of the

student. Let's let J be 1 and the values of the variables be as follows:

```
N$(1)="Jane Smith": G(1,1)=95: G(2,1)= 84: G(3,1)=78
```

Then when the results are printed we would have:

```
Jane Smith  95  84  78
```

This is the way some teachers handle their grade book. What we want to do is to show how to handle the information and how to use arrays. Let's use the first method and carry the variables within our statements. We need to cover another NEW concept before we continue .

FOR - NEXT LOOPS

When the same statemnts are to be executed a number of times then FOR-NEXT LOOPS can be used. The format is as follows:

```
10 FOR J = 1 to 10
20 ?"HELLO"
30 NEXT J
```

When this program is run the word "HELLO" will be printed 10 times. Statement 10 is the beginning and tells how many times to do the task. All of the statements are executed until statement 30 or a "NEXT J" command appears. To prin the names of the students in our array we could insert for statement 20

```
20 ?N$(J)
```

When J=1 the name of the first student will be printed and when J=2 the name of the second student will be printed. To show how this is done we are including a grade book program. It is for 5 students and 3 grades but can be expanded by changing the numbers in the DIM statement.

GRADE BOOK PROGRAM

This program will print the names of 5 students and their grades. This is an example of using arrays and the FOR- NEXT loops. The variables are contained within the statements. Notice the simple print command in the FOR-NEXT loop.

```
5 ?"GRADE BOOK PROGRAM
10 ?"PROGRAM # 6-1-85
15 ?"COPYRIGHT (c) 1985
20 ?"DYNAMIC ELECTRONICS INC.
30 'NOTE THE ENDING QUOTATION
35 'MARKS ARE NOT REQUIRED IF
40 'A PRINT STRING IS THE LAST
45 'PART OF A STATEMENT
50 DIM N$(5), G(3,5)
55 '5 STUDENTS, 15 GRADES
60 N$(1)="SMITH":G(1,1)=95
65 G(2,1)=73: G(3,1)=68
70 N$(2)="GREENE":G(1,2)=87
75 G(2,2)=73:G(2,3)=79
80 N$(3)="WALLACE":G(1,3)=56
85 G(2,3)=72: G(3,3)=81
90 N$(4)="YOUNG":G(1,4)=95
95 G(2,4)=91: G(3,4)=85
100 N$(5)="YOGA":G(1,5)=72
105 G(2,5)=65: G(3,5)=79
110 'PRINT THE DATA
115 FOR J=1 TO 5
120 ?N$(J),G(1,J);G(2,J);G(3,J)
125 NEXT J
130 'NOTE J REPRESENTS THE
135 'NUMBER OF THE STUDENT
```

COMPUTER GRAPHICS

PART 5

In preceeding issues on this series we showed the principles of Color Computer graphics. We discussed the various semigraphic and graphic modes and showed how they could be initialized by various memory pokes. Last month we shouwed how to use some extended basic commands for setting up the graphics modes.

Rather than take on new material this month we want to show how to use the concepts covered. As an example we are including a

graphics program that allows characters to be drawn on the screen as you type them in from the keyboard. If you will recall the graphics elements are displayed across the screen with one bit controlling a graphics element or pixel. Therefore an 8 bit byte controls 8 graphic elements in the high resolution graphics modes.

CHARACTER GENERATION

Characters are generated by turning dots on or off in the proper order. A standard format is an array of dots arranged as 5h by 7v. The information for the character could be contained in 5 bytes. However we will use 7 bytes because it is easier to implement. To draw a character we will need to write the value of the first row into memory. Then each of the remaining 6 rows are written into memory. We have to add a value equal to the number of characters on each row so that we can move down the screen vertically.

At the end of a row, the memory location has to be increased by a value equal to the number of rows multiplied by the number of characters per row. A counter can keep track of the number of characters displayed on a row and the program can increase the memory by the correct amount so that the first character on the next row will be in the correct position.

Our example program has two sizes for the characters. One size gives 16 characters per line. For these larger characters we wrote the same character information in two vertical locations. Therefore it takes 14 bytes for each character. This gives dark easy to read characters. The smaller size characters allow 32 characters across the screen which is similar to the normal power up display.

DATA APPROACH

There are several options for handling the data. We could use READ and DATA statements, we could have a seperate machine language program for data, we could carry the data within remark statements, or we could define the data within the program. Since memory was not a problem we decided to use the latter approach and define the data within the program. We decided to use an array called K$(N) and let N be the ASCII value of the character. We carried each of the 7 bytes within the K$(N) string as 2 character decimal values. This made each K$(N) string 14 characters long. To remove each byte we used the MID$ basic command.

Another advantage of this approach is that we can add additional characters to our set as they are needed. With the demonstration program only capital letters, numbers and some punctuation are included. The complete 128 character set could be added by defining the characters in additional K$(N) statements.

The principles we are covering for using graphics with a television or monitor also apply to dot matrix printers. In the graphics mode dot matrix printers allow a byte to define 7 or 8 dots depending upon the printer. So the techniques for doing printer graphics are similar to those presented here.

There is much more to be covered in this series. We want to show how to plot data and make graphs. With our knowlege of character generation presented in this issue we will be able to label our graphs and charts.

+ ++ + ++ + ++ ++ ++ + ++ + ++ +

RENEWAL TIME?

The date beside your name on the address label indcates the last issue you will receive. Send in your renewal if you want to continue receiving technical information on Color Computers. This is the last issue for those with 6/85.

+ ++ + ++ + ++ ++ ++ + ++ + ++ +

GRAPHICS DEMONSTRATION PROGRAM
(CHARACTER GENERATOR)

```
5 'EXTENDED BASIS IS REQUIRED
10 CLS
20 PRINT"THIS IS PGM 6-2-85
30 PRINT"COPYRIGHT (c) 1985
40 ?"DYNAMIC ELECTRONICS INC.
50 PRINT
60 'RESERVE ROOM FOR GRAPHICS
70 PCLEAR 8
80 INPUT"ENTER 1 FOR DISK
   DRIVE";D
90 'SET UP A CHARACTER ARRAY
100 DIM K$(100)
110 M=3072
120 IF D=1 THEN M=5120
130 'LET M=5120 FOR DISK DRIVE
140 INPUT"2 FOR SMALL CHAR OR 1
    FOR LARGE CHAR";P:P=2*P
150 'GO TO GRAPHICS MODE
160 PMODE P,2: SCREEN 1,1
170 PRINT"ENTER A CHARACTER
180 '
190 'CHARACTER DATA FOLLOWS
200 'THE 7 ROWS ARE DEFINED BY
210 'THE DECIMAL VALUES OF THE
220 'PAIRS OF CHARACTERS
230 'OTHER CHAR CAN BE ADDED
240 K$(42)="04211404142104
250 K$(43)="00040431040400
260 K$(44)="00000000040408
270 K$(45)="00000031000000
280 K$(46)="00000000000004
290 K$(48)="14171921251714
300 K$(49)="04120404040414
310 K$(50)="14170106081631
320 K$(51)="31010206011714
330 K$(52)="02061218310202
340 K$(53)="31163001011714
350 '54=6
360 K$(54)="07081630171714
370 K$(55)="31010204080808
380 K$(56)="14171714171714
390 K$(57)="14171715010228
400 K$(58)="00000400040000
410 K$(59)="00000400040408
420 K$(60)="02040816080402
430 K$(61)="00003100310000
440 K$(62)="08040201020408
450 K$(63)="14170204040004
460 K$(65)="04101717311717
470 K$(66)="30171730171730
480 K$(67)="14171616161714
490 K$(68)="30171717171730
500 K$(69)="31161630161631
510 K$(70)="31161630161616
520 K$(71)="15161616191715
530 K$(72)="17171731171717
540 '73=I
550 K$(73)="14040404040414
560 K$(74)="01010101011714
570 K$(75)="17182024201817
580 K$(76)="16161616161631
590 K$(77)="17272121171717
600 K$(78)="17172521191717
```

```
610 K$(79)="14171717171714
620 K$(80)="30171730161616
630 K$(81)="30171717211813
640 '82=R
650 K$(82)="30171730201817
660 K$(83)="14171614011714
670 K$(84)="31040404040404
680 K$(85)="17171717171714
690 K$(86)="17171717171004
700 K$(87)="17171721212717
710 '88=X
720 K$(88)="17171004101717
730 K$(89)="17171004040404
740 K$(90)="31010204081631
750 PMODE P,2:SCREEN 1,1
760 C=0:G=P/2:H=16*G
770 'THE FOLLOWING MAKES THE
780 'COMPUTER WAIT FOR YOU.
790 A$=INKEY$: IF A$="" THEN
    790
800 'CONVERT THE STRING TO A
810 'VALUE
820 A=ASC (A$):N=M
830 'CHECK FOR LEFT ARROW
840 IF A=8 THEN M=M-1:C=C-1:GO
    TO 790
850 '
860 'BREAK THE STRING K$ DOWN
870 'TO THE SEVEN PARTS FOR THE
880 'CHARACTER ROW INFORMATION
890 FOR J=1 TO 7
900 Q=2*J-1:X$=MID$(K$(A),Q,2):
    X=255-VAL(X$)
910 'DOUBLE WRITE FOR LARGE CH
920 IF P=2 THEN POKE N+16,X
930 POKE N,X:N=N+32
940 NEXT J
950 'LEAVE SPACE BELOW CH
960 POKE N,255: POKE N+32,255
970 M=M+1:C=C+1
980 'CHECK FOR END OF LINE
990 IF C=H THEN M=N+1: GO TO
     750
1000 GO TO 790
```

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

LARGE MEMORY PROGRAMS
(PART 5)

In this series we have shown how to initialize the second 32K memory bank for 64K computers. We showed how to copy the first 32K bank into the second 32K bank so that basic can be run in both banks. We also showed how to exchange the contents of both banks so that you can run the program in the hidden 32K bank and save the programs in the normal 32K bank. Also we showed how to start a basic program in one bank and continue it into the second bank. This works fine except the variables are not transferred from one bank to the other. For large programs we can divide them up into two relatively equal halves and put one half in one bank and the other in the second bank. Then depending upon what is selected from the menue, will determine which bank is used.

DATA STORAGE in BANK 2

There are applications where it is desireable to store data in the second bank. Examples are address files, check book entries, invoice data, inventory, etc. In this issue we are including a program that allows you to enter information and store it in the second bank. You can review the information by printing it to the screen and you can add new information. We called it an address file but it could used for other applications. There is more that could be done to the program, but it will demonstrate the principle of storing information in the second bank.

To do this we need a machine language subroutine link. This is included within our program in DATA statements. We reserved a memory area from 4015 to 4200 for our machine language subroutine links. Data can reside anywhere within the second bank but must skip around our machine language subroutines. We started our data at 4200 to keep the program simple. Also we used locations 498-500 for temporary memory storage. We used locations 474-479 for the vector locations for transferring data. If you understand what we did in the program then you should be able to write your own large memory program.

ADDRESS FILE PROGRAM
(2-BANK DEMO PROGRAM)

```
2 'EXTENDED BASIC IS REQUIRED
5 PCLEAR 8
10 IF PEEK (4015) =142 THEN 20
   ELSE GO SUB 6000
20 PRINT"THIS IS PROGRAM 6-3-85
30 PRINT"2 BANK ADDRESS FILE
40 PRINT"DEMONSTRATION PROGRAM
50 PRINT"COPYRIGHT (c) 1985
60 PRINT"DYNAMIC ELECTRONICS
   INC.
70 '498 AND 499 - END OF DATA
    VECTOR
80 '500 BYTE BEING TRANSFERRED
90 PRINT:PRINT"1 COMPOSE NEW
   ADDRESS FILE
100 PRINT"2 PRINT ADDRESSES TO
    SCREEN
110 INPUTX
120 ON X GO SUB 200,400
130 GO TO 90
200 CLS:PRINT"THIS COMPOSES AN
    ADDRESS FILE
210 PRINT"USE RIGHT ARROW KEY
    TO EXIT THIS SECTION
220 PRINT"START ENTERING
    INFORMATION
230 PRINT"DATA IS STORED IN THE
    OTHER BANK STARTING AT 4200
240 PRINT"WE ARE USING 500 FOR
    TEMPORARY CHARACTER STORAGE
250 INPUT"ENTER A TO ADD TO
    FILE";K$
260 IF K$="A" THEN
    X=256*PEEK(498)+PEEK(499)
    ELSE X=4200
270 GO SUB 380:POKE 478,MS:
```

```
    POKE479,LS
280 M=X
290 X=500:GO SUB 380: POKE474,
    MS:POKE476,MS:POKE475,LS:
    POKE477,LS
300 A$=INKEY$:IF A$=""THEN 300
310 A=ASC(A$):IF A=8 THEN
    M=M-1:GO TO 300
320 X=M:GO SUB 380
330 IF A=9 THEN POKE 498,MS:
    POKE 499,LS: PRINT"FILE
    ENDS AT"X:RETURN
340 'DESIGNATE MEMORY IN 2ND
    BANK
350 POKE 478,MS:POKE479,LS
360 POKE500,A :PRINTA$;
370 EXEC 4090: M=M+1: GO TO 300
380 MS=INT(X/256): LS=X-256*MS
390 RETURN
400 PRINT"THIS PRINTS THE
    ADDRESSES TO THE SCREEN.
410 'SET UP VECTORS FOR DATA
    TRANSFER
420 M=4200
430 X=500:GO SUB 380: POKE
    478,MS: POKE479,LS
440 X=M:GO SUB 380: POKE 474,
    MS: POKE 475,LS: POKE 476,
    MS: POKE477,LS
450 EXEC 4115: M=M+1:
    A=PLEK(500): A$=CHR$(A)
460 PRINTA$;
470 IF A>127 THEN ?: ?"THIS IS
    THE END OF THE FILE.": POKE
    498,MS: POKE499,LS: ?"FILE
    ENDS AT"X: GO TO 90
480 GO TO 440
490 PRINT"THIS LOOKS AT VALUES
    IN MEMORY
500 INPUT"MEMORY";M
510 A=PEEK(M):A$=CHR$(A)
520 V=256*A+PEEK(M+1)
530 X$=INKEY$:IF X$="M"THEN
    INPUT"MEMORY";M
540 PRINTM;A;A$;V
550 M=M+1: GO TO 510
6000 PRINT"BANK SWITCHING
     SUBROUTINES
6010 FOR J = 4015 TO 4138
6020 READ X: POKE J,X
6030 PRINTJ;X
6040 NEXT J
6050 EXEC 4015: EXEC 4070
6060 PRINT"THE SECOND BANK IS
     READY.
6070 PRINT"EXEC 4090 FOR B0-B1
     DATA TRANSFER
6080 PRINT"EXEC 4115 FOR B1-B0
     DATA TRANSFER
6090 PRINT"BEG VECTOR IN 474
6100 PRINT"END VECTOR IN 476
6110 ?"NEW LOCATION VECTOR IN
     478
6120 RETURN
6130 PRINT'BANK INITIALIZATION
6140 DATA 142, 15, 204, 16,
     142, 143, 204, 166, 128,
     183, 255, 214, 183, 255
6150 DATA 223, 167, 160, 183,
     255, 222, 140, 17, 148,
     45, 238, 57, 18, 18, 18
6160 'BANK EXCHANGE  (4044)
6170 DATA 79, 95, 31, 1, 166,
     132, 183, 255, 213, 230,
     132, 167, 132, 183
6180 DATA 255, 212, 231, 128,
     140, 127, 255, 35, 237,
     57, 18, 18
6190 'COPY B0 TO B1 (4070)
6200 DATA 79, 95, 31, 1, 166,
     132, 183, 255, 213, 167,
     128, 183, 255
6210 DATA 212, 140, 127, 255,
     35, 241, 57
6220 'B0 TO B1 DATA  (4090)
6230 DATA 190, 1, 218, 16, 190,
     1, 222, 166, 128, 183,
     255, 213, 167, 160
6240 DATA 183, 255, 212, 188,1,
     220, 35, 241, 57, 18, 18
6250 'B1 TO B0 DATA  (4115)
6260 DATA 190, 1, 218, 16, 190,
     1, 222, 183, 255, 213,
     166, 128, 183, 255
6270 DATA 212, 167, 160, 188,
     1, 220, 35, 241, 57,0
```

NEW PRODUCTS

This section is available free for producers and dealers of color computer products. If you would like your new product listed here send a description of the product to:

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

OS-9 SOLUTION

Spectrum Projects, Inc. has announced the release of its first entry into the OS-9 marketplace with a front-end user interface software system designed to make OS-9 user-friendly and completely menu-drived.

Designated the OS-9 SOLUTION, it replaces nineteen OS-9 commands with single keystroke, menu-driven, easy to use operations. By using a directory window with the ! and up arrow keys for access and command execution, it allows multiple copying, killing and info printouts for whole or partial directories. All XMODE parameters can be set at the touch of a single key and the need to type in complex long pathnames has now been eliminated. The OS-9 SOLUTION costs $39.95 and requires OS-9 version 01.01.00.

THUNDER RAM

Spectrum Projects, Inc. has announced the release of the first 256K memory upgrade board for the Tandy/Radio Shack Color Computer.

Designated THUNDER RAM, it has the following features: Emulation of a 40 track RAMDISK, up to 30 times faster than an ordinary floppy disk drive, a full 60K Print Spooler, create Basic programs up to 128K long or store up to 30 or more Hi-Res screens in memory at one time.

THUNDER RAM costs $119.95 and requires a Color Computer and 1 Disk System.

For more information on either of these products contact Spectrum Projects Inc.; 93-15 86th Drive; P. O. Box 21272; Woodhaven, NY 11421

DYHEAT

Dynamic Electronics Inc. is proud to announce a solution to the heat problem for D, E, and 285 type Color Computers. DYHEAT consists of a 5 volt regulator mounted to a 6 inch aluminum heat sink which mounts outside the computer. This replaces the normal 5 volt regular within the computer removing the major cause of heat.

Excessive heat within the computer can cause component stress and failures of the expensive integrated circuits. DYHEAT is easy to install. One resistor has to be cut, one wire soldered, and two wires connected to the 5 volt and ground pins by sliding eyelets over appropriate leads of one of the integrated circuits. Two holes have to be drilled into the case of the computer for mounting the aluminum heat sink. DYHEAT costs $19.95.

For more information contact Dynamic Electronics Inc.; P. O. Box 896; Hartselle, AL 35640; (205) 773-2758

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.

X-RATED Adventure Game! A game so real that you will know that your COCO has finally grown up. 16/32K $15 tape/ $20 disk; Y. Kraft, 164 N. Blackstone Ave., Fresno, CA 93701

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

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

****
* Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I *
* Want to receive instruction on programming, Computer Theory, Operat- *
* ing 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

BULKRATE
U.S. POSTAGE
PAID
HARTSELLE, AL
35640
PERMIT NO. 21