ENGINEERING NOTES
on
Radio Shack Color Computers

MAY 1985
Vol. 2 No. 4

$1.95

```
****    *   *   *   *     *     *   *    ***     ***
*   *   *   *   *   *    * *    ** **     *     *   *
*   *    * *    **  *   *   *   * * *     *     *
*   *     *     * * *   *****   * * *     *     *
*   *     *     *  **   *   *   *   *     *     *
*   *     *     *   *   *   *   *   *     *     *   *
****      *     *   *   *   *   *   *    ***     ***
```

```
 ***     ***    *        ***    ****
*   *   *   *   *       *   *   *   *
*       *   *   *       *   *   *   *
*       *   *   *       *   *   ****
*       *   *   *       *   *   *  *
*   *   *   *   *   *   *   *   *   *
 ***     ***    *****    ***    *   *
```

```
*   *   *****   *   *    ***
*   *   *       *   *   *   *
**  *   *       *   *   *
* * *   ***     * * *    ***
*  **   *       * * *       *
*   *   *       ** **   *   *
*   *   *****   *   *    ***
```

* GAS MILEAGE PROGRAM
* GRAPHICS DEMO PROGRAM
* DATA MOVE PROGRAM

* LARGE MEMORY PROGRAMS (Part 4)
* WRITING PROGRAMS (Part 3)
* BASIC PROGRAMMING

* COMPUTER GRAPHICS (Part 4)
* 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

May 1985

Editor and Publisher
Bill Chapple

Secretary
Deanne Hill

CONTENTS

* 96K-M EXPANDER *

You have a 64K computer but can only use 32K. Our 96KX-M module allows full use of both 32K memory banks. Run BASIC in both, transfer data from one bank to the other, or continue a BASIC program into the other bank. Nothing to load just EXEC 57701 when you need the software. Does not use any of your computer's RAM. Powerful utilities also included. 96KX-M module mounts inside. $59.95
96KX-C plug in cartridge. $49.95

* VIDEO REVERSER *

Provides (1) Reversed, (2) Reversed all capitals, & (3) Normal. Solderless installation. VR-1 $19.95
VR-1M Plug in module mounts under VDG chip. $24.95

64K MEMORY EXPANDERS

No trace cutting, solderless, & reversible.

ME-4	. . . D & E computers to 64K	$89.95
ME-4F	. . F or 285 computers to 64K	$79.95

128K EXPANDERS

Plug-in modules mount inside your computer. Compatible with all software. Transfer variables & program control from one 64K bank to the other.

ME-128-64	Upgrades 64K computers to 128K	$129
ME-128D	Upgrades D & E computers to 128K	$179
ME-128F	Upgrades F or 285 computers to 128K	$189
ME-5	Set of 8-64K chips (PRICE REDUCED)	29.95

DISASM - Decimal 6809 Assembler-Disassembler	19.95
DYTERM - 300 to 2400 baud Terminal Program	$14.95
MPM - Stack 5 programs in your computer	$14.95

UNINTERRUPTED POWER SOURCE (UPS) new - Save your Programs during power failures. Battery & Electronic circuit mount under keyboard. UPS $59.95

ADD A SECOND PORT (PORT-2)

A plug in module with a switch to add a second port. Requires cutting one resistor and soldering 2 wires. Runs most cartridge programs. Gold plated contacts for long life. PORT-2 $29.95

* DEALER & CLUB INQUIRIES * * FREE CATALOG *

24 hr phone. Checks, VISA & MC cards. Add $2 ship.

DYNAMIC ELECTRONICS INC.
Box 896 (205) 773-2758
HARTSELLE, AL 35640

EDITOR'S COMMENTS

One of the hardest things I have had to deal with with Color Computers is the heat problem. I have drilled holes in covers, operated computers without covers, and tried various other techniques for reducing the heat build up in the computers.

In case you don't understand why the heat builds up let me briefly explain what happens. The memory chips require 5 volts. This voltage is derrived from an unregulated 9 to 10 volt source which consists of the power transformer, rectifiers, and a large filter capacitor. A series voltage regulator reduces the voltage to a precise 5 volts. The regulator is mounted on a large heat sink on the circuit board. The regulator absorbs the excessive voltage and generates heat due to the current flowing through it. The electronic design is very good. None of our computers have failed due to a faulty regulator. To better understand the power consumed by the regulator suppose the following situation occurs. If the input to the regulator is 10 volts and the output is 5 volts, then the regulator has 5 volts applied across it and consumes the same power as the 5 volt electronic chips. When you add extra accessories then you cause more power to be required of the regulated thus generating more heat.

If the commercial line voltage increases or decreases then the regulator absorbs the change in voltage keeping the output at the constant 5 volt level.

Since the electronic design is good, the only problem is "how do we get the heat out of the computer"? We can use a fan which is bulky and creates noise and possible electical problems. I decided that what needs to be done is to move the regulator outside the case. I took a large heat sink with fins and replaced the regulator with another one mounted to the external heat sink. The results are very gratifying. The computer stays cool and I haven't had any heat related problems. Next month we will show how you can modify your computer to reduce the internal heat.

WRITING PROGRAMS
(Part 3)

For the past two months we have been introducing some of the fundamentals of Basic Programming. We talked about statement numbers, remarks, PRINT commands, LIST commands, the NEW command, and the Clear Screen (CLS) command. Our purpose in these series is to teach those who do not have any programming experience how to write useful Basic Programs.

If you will follow the examples we give and understand the demonstration programs then you are on the way to writing your own programs. Of course the advantage of writing your own programs is that you can customize the program to suit your individual needs. A computer without software is not of much use. Software can be purchased or written. You can purchase software that will do a specified job. However only you know what your tasks are and what you want the program to do. You can hire a programmer to write your programs which is expensive, or you can learn to write them yourself. So our objective is to help you to develop programming skills so that you can write your own programs.

Using Variables

Last month we talked about variables and showed the format for entering variables into the computer. Also we demonstrated printing variables. This month we want to show how you can use variables in a program. There are several ways a variable can be entered into a program. One way is to define the variable within Basic statements. This is what we did last month with our Print Demonstration Program. Another method is to use the "INPUT" command.

INPUT Command

The input command is used to enter data into the computer as the program is running. The computer stops when it encounters an input command and waits until you enter the variable. When the computer stops a "?" mark will appear on the screen. You then type in the variable and press the <ENTER> key. The computer then continues. If several variables are entered during the program execution, it is advisable to have the name of the variable printed with the "?". This can be done with a print statement or included within the input command.

Suppose we let the variable "D" represent distance. Then we cam write an input command as follows:

```
 30 INPUT "DISTANCE";D
```

The characters within the parenthesis will be printed followed by the question mark.

Another way to do this is to use a print statement before the input command. The following illustrates this.

```
 10 ?"DISTANCE";
 20 INPUT D
```

Remember we talked about different print formats last month. The ";" in Statement 10 means to move the cursor one position after the print command. A "," means to move to the next field and a blank means to move down to the next line. The short notation in statement 30 is preferred because it takes less memory and is easier to type.

The procedure is the same for entering string variables from the keyboard. Suppose we want to enter a player's name on a ball team. Then the following statement is an example of how to do this:

```
 50 INPUT "PLAYER'S NAME"; N$
```

Notice the $ after N to designate a string variable.

OPERATORS

Operators allow mathematical operations to be performed between two numerical variables. For basic arithmetic operations we need to add, subtract, multiply, and divide. The operators for these operations are

 + for addition
 - for subtraction
 * for multiplication
 / for division

Suppose we want to add two variables X and Y. We have to call the sum a new variable. Let's use Z for the new variable. Now we can write

```
 30 INPUT "ENTER X";X
 40 INPUT "ENTER Y";Y
 50 Z = X + Y
 60 ?"THE SUM IS";Z
```

The preceeding program allows the variables to be entered and the sum printed on the screen in statement 60. You might want to enter this program to become familiar with the INPUT command.

GAS MILEAGE PROGRAM

This useful program will allow you to calculate the miles per gallon for your automobile, truck, or motorcycle. It uses the principles covered in this series. The procedure is to enter the beginning miles "B", the ending miles "E", and the gallons used "G". Let's use "M" for miles per gallon. Then the following formulas will calculate M.

$$A = E - B$$

$$M = A / G$$

The following is a listing of the program with comments.

```
 10 ?"GAS MILEAGE PROGRAM"
 20 INPUT "BEGINNING";B
 30 INPUT "ENDING";E
 35 INPUT "GALLONS";G
 40 'CALCULATE MILES TRAVELLED
 50 A = E - B
 60 'CALCULATE GAS MILEAGE
 70 M = A / G
 75 'PRINT THE ANSWER
 80 ?"MPG = "M
 90 GO TO 20
```

LARGE MEMORY PROGRAMS
(Part 4)

In the preceeding editorials on this series we showed how to configure the computer to use either of the 32K memory banks. Our procedure was to copy each byte from the first bank to the second bank. The first bank is initialized for Basic operation when the computer is turned on, and since each byte was copied from the first bank into the second bank we could run Basic in the second bank too. We also gave machine language subroutines that allowed the exchanging of information in the two banks. This means that you can load a program in the first bank and copy the first bank to the second bank. Then you can load a different program in the first bank. You can run the second program, edit it, and at any time you can exchange it for the first program.

The information in the second bank is completely protected. If the computer hangs up, or you perform a master reset, then the information in the second bank is not altered. This is a very nice feature especially if you are working on a program that has a few bugs in it. Before running the program make a copy of it in the second bank. Then if you have to reset the computer you can recall the program by exchanging the information in the banks and then you can again copy the program into the second bank.

Last month we showed how to extend a basic program from one bank into the other. This works very well but you have to remember that variables are not transferred from one bank to the other. If it is desireable to have a long program in two banks then an easy way to organize it would be to put about half in one bank and half in the other bank. Suppose you have two menus with different options you want to select. Then one menu with its options could be in one bank and the second menu could be in the second bank. This approach does not require much software since the Basic that comes with the computer is operating in both banks. To link the banks together then let a bank switch routine occur if the <ENTER> key is pressed. We used a machine language subroutine at 4044 to do the bank switching. To call this from Basic just EXEC 4044.

DATA STORAGE

Suppose it is desireable to store data in the second bank. An example for this requirement would be an address file. If each address took 100 bytes then we could store 320 addresses in a 32K memory bank. We want to show how data can be stored in the second bank. Let's look at what we will have to do to accomplish this task.

1. We will need a vector to show the beginning and ending of the data plus another vector to show where the data is going. (See Vol.1 No. 1 for a discussion of vectors)

2. We will load an index register (X) with the beginning of data in the first bank and another index register (Y) with the location of data in the second bank.

3. Next we will load the "A" register with one byte of data at the location designated by X, switch banks, and store the data at the location designated by "Y".

4. We will come back to the first bank, get another byte, and continue the process until we have finished.

Vector Memory Locations

Memory locations that are not critical to the computer's operation have to be used for the vectors. Locations from 474 to 480 are used for the name of a cassette file and can also be used for our purpose. So we will use the following locations for our vectors.

474 - Beginning of data
476 - End of data
478 - Location in bank 1

We will need two machine language subroutines. The first will transfer data from the first bank to the second bank. The second subroutine will return the data from the second bank to the first bank.

B0 - B1 Block Move

The following is an assembly listing of a data block move from bamk 0 to bank 1. All numbers are in decimal.

```
4090 LDX E 474 'PUT BEG.
     VECTOR IN X
4093 LDY E 478 'PUT END.
     VECTOR IN Y
4097 LDA X DIR R+ ' LOAD
     A INDEXED WITH X AND
     INCREMENT X
4100 STA E 65493 ' SWITCH
     TO BANK 1
4102 STA Y DIR R+ 'STORE
     A INDEXED WITH Y AND
     INCREMENT Y
4105 STA E 65492 'SWITCH
     TO BANK 0
4107 CMPX E  476 'IS X
     =476 VECTOR?
4110 BLS 4100 'GO TO 4100
     IF X<= THE VALUE IN 476
4112 RTS 'RETURN FROM SUB
```

B1 - B0 Block Move

The following assembly listing is for block moving from bank 1 to bank 0.

```
4115 LDX E 474 'X=BEG
4118 LDY E 478 'Y=END
4122 STA E 65493 'SWITCH TO
     BANK 1
4125 LDA X DIR R+ 'LOAD A
     INDEXED TO X AND
     INCREMENT X
4127 STA E 65492 'SWITCH TO
     BANK 2
4130 STA Y DIR R+ 'STORE A
     INDEXED TO Y AND
     INCREMENT Y
4132 CMPX E 476 'ARE WE
     THROUGH?
4135 BLS  4127 'GO TO 4127 IF
     X <= THE VALUE IN 476
4137 RTS
```

Our assembly listings are in decimal so those who are not familar with assemblers can see what we are doing. Since basic uses decimal arithmetic we feel that this is less confusing to readers who are unfamilar with assemblers. We are going to give a basic program that you can use to enter the subroutines. Then you can write your own basic program and use the machine language subroutines for bank switching and moving the data from one bank to the other.

To use the subroutines remember that they must occupy the same memory areas in both memory banks. We chose 4000 - 4200 as our subroutine area so this area needs to be protected in both memory banks. Also when you are setting up the vectors in locations 474 to 478 the most significant (MS) byte goes in the lower location and the least significant (LS) byte goes in the upper location. Formulas for breaking a vector into its two byte components are as follows:

```
 MS = INT (V / 256)
 LS = V - 256 * MS
```

To write a program that stores data in the second bank, it is necessary to EXEC 4015 and then EXEC 4070. This initializes the second bank for basic. Then EXEC 4090 to move a block of data from bank 0 to bank 1 with the vectors in 474-479. EXEC 4115 moves a block from bank 1 to bank 0. Next month we will give you an example program for storing data in the second bank.

DATA MOVE PROGRAM

The following basic program allows the machine language subroutines discussed in this series to be loaded into 4000 - 4200 in memory. You can save this program on a cassette or disk and reload the machine language subroutines whenever you need them.

BANK SWITCHING
Program Listing

```
1 'PGM # 5-1-85
5 PCLEAR 8
10 ?"BANK SWITCHING SUBROUTINES
15 ?"COPYRITE (c) 1985
20 ?"DYNAMIC ELECTRONICS INC.
25 FOR J = 4015 TO 4138
```

```
30 READ X: POKE J,X
35 ?J;X
40 NEXT J
45 EXEC 4015: EXEC 4070
50 ?"THE SECOND BANK IS READY.
60 ?"EXEC 4090 FOR P0-P1 DATA
     TRANSFER
70 ?"EXEC 4115 FOR P1-P0 DATA
     TRANSFER
75 ?"BEG VECTOR IN 474
80 ?"END VECTOR IN 476
85 ?"NEW LOCATION VECTOR IN
     478
90 ?'BANK INITIALIZATION DATA
100 DATA 142,15,204,16,142,143,
    204,166,128,183,255,167,255
110 DATA 223,167,160,183,255,
    222,140,17,148,45,238,57,18,
    18,18
115 'BANK EXCHANGE  (4044)
120 DATA 79,95,31,1,166,132,183,
    255,213,230,132,167,132,183
130 DATA 255,212,231,128,140,
    127,255,35,237,57,18,18
135 'COPY B0 TO B1 (4070)
140 DATA 79,95,31,1,166,132,
    183,255,213,167,128,183,255
150 DATA 212,140,127,255,35,241,
    57
155 'B0 TO B1 DATA  (4090)
160 DATA 190,1,218,16,190,1,222,
    166,128,183,255,213,167,160
170 DATA 183,255,212,188,1,220,
    35,241,57,18,18
175 'B1 TO B0 DATA  (4115)
180 DATA 190,1,218,16,190,1,222,
    183,255,213,166,128,183,255
190 DATA 212,167,160,188,1,220,
    35,241,57,0
```

COMPUTER GRAPHICS

(Part 4)

For the past three editorials in this series we have showed how to use graphics directly by various memory pokes. This month we want to use a few extended basic commands and show how to use the highest resolution graphics. Let's define a few extended basic commands that we will be using.

PCLEAR X

This command reserves X number of graphisc pages. X can be any number from 1 to 8. This should be the first statement in your program or it should be executed from the keyboard before a basic program is loaded. Your basic program is moved to a different memory location when this command is used. Graphics pages always preceed your basic program in memory.

PMODE M,P

This sets up the graphics modes. M is the graphics mode number and P is the graphics page.

SCREEN T,S

For graphics T must be 1. S is the color value which can be either 0 or 1.

GRAPHICS PAGES

A graphics page requires 1536 bytes of memory. The first graphics page is from 1536 to 3072. We will be using this graphics page for our discussion and programs this month. Because this area conflicts with the RAM required for a disk drive, our example program will not work with a disk drive.

Let's look at how the graphic information is carried within a byte. Each bit of the byte controls a graphic element. If the bit is a "1" the the element is "ON and if it is a "0" the element is "OFF".

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

If the bits are numbered from 0 to 7, then the weight of each bit is as shown. If all elements are to be on then the value of the byte is 255. If you want elements 3 and 4 to be on then the value of the byte is 8 + 16 or 24. The elements are displayed horizontally across the screen with the most significant bit (7) being displayed first or to the left of the others. There are 32 bytes across the screen and this multiplied by 8 gives 256 horizontal elements. One page is made up of 192 vertical elements. The 192 * 32 product gives 6144 bytes which is required for the highest resolution graphics. This occupies four 1536 byte graphics pages.

We developed a graphics demonstration program so that you can write to each individual byte or write the same value in each location of page 1. The first part allows writing the same value to each location or by using the random number generator, random values can be written to the locations. Also you can copy page 1 into any of the other 4 graphics pages.

The second part of the program allows you to enter a value and by using the arrows draw a pattern on the screen. We also included a feature that allows you to move the cursor without changing the value as you cross a memory location to keep from altering your pattern. The program has instructions with it so you should not have any problems in running it.

RENEWAL TIME?

The date beside your name on the address label indicates the last issue you will receive. Send in your renewal if you want to continue receiving technical information Color Computers. This is the last issue for those with 5/85.

GRAPHICS DEMO PGM

```
10 'THIS IS PGM 5-2-85
20 CLS
30 ?"GRAPHICS DEMONSTRATION PGM
40 PRINT"COPYRITE (c) 1985
50 ?"DYNAMIC ELECTRONICS INC.
60 PRINT
70 INPUT"ENTER 1 FOR DRAWING ON
   THE SCREEN OR PRESS ENTER TO
   CONTINUE";W
80 IF W=1 THEN 400
90 INPUT"GRAPHIC PAGES TO
   CLEAR  0-4";M
100 PRINT" TO ALTER THE PROCESS
110 PRINT"ENTER A 'C' TO COPY
    GRAPHICS PAGE 0
120 PRINT"ENTER A 'W' TO WRITE
    TO EACH LOCATION
130 PRINT"ENTER A 'P' TO CHANGE
    GRAPHICS PAGES.DISPLAYED
140 INPUT "R FOR RANDOM
    NUMBERS";Y$
150 IF Y$="R" THEN 170
160 INPUT "VALUE";A
170 B=1536: E=2*B-1
180 'PMODE M,P
190 'M=GRAPHICS MODE & P= PAGE
200 PMODE M,1
210 'SCREEN T,S : IF T=0 THEN
220 'TEXT IF T=1 THEN GRAPHICS
230 'FOR S A VALUE OF 0 OR 1
240 'SELECTS A COLOR SET.
250 SCREEN 1,1
260 FOR J=B TO E
270 SCREEN 1,1
280 IF Y$="R" THEN A=RND(255)
```

```
290 X$=INKEY$:IF X$="" THEN 300
    ELSE INPUT"VALUE";A
300 POKE J,A: NEXT J
310 Y$=INKEY$: IF Y$="" THEN
    310
320 IF Y$="C" THEN INPUT"PAGE
    NUMBER";X: PCOPY 1 TO X:
    PMODE M,1:SCREEN 1,1: GO TO
    310
330 IF Y$="W" THEN 400
340 IF Y$="P" THEN INPUT"NUMBER
    OF GRAPHICS PAGES TO
    DISPLAY";N:PMODE N,1:SCREEN
    1,1: GO TO 310
350 GO TO 90
400 'THIS WRITES VALUES TO
    INDIVIDUAL BLOCKS
410 INPUT"MEMORY - BEGINNING IS
    1536. PRESS ENTER FOR
    1536";Z
420 IF Z=0 THEN Z=1536
430 PRINT"MEMORY="Z;" VALUE="V:
    INPUT"ENTER VALUE";V
440 IF Z<1536 THEN ?"MEMORY TOO
    LOW. RESETTING MEMORY TO
    1536":Z=1536
450 POKE Z,V
460 PMODE 1,1:SCREEN 1,1
470 PRINT"USE THE ARROWS FOR
    POSITIONING
480 PRINT"ENTER KEY RETURNS TO
    BEGINNING OF THE LINE
490 PRINT"'R' KEY RETURNS TO
    THE MAIN PGM
500 PRINT"'V' ALLOWS CHANGING
    VALUES (0-255)
510 PRINT"'W' WRITES VALUE TO
    MEMORY
520 PRINT"'P' MOVES CURSOR ONLY
530 PRINT"'I' RETURNS TO THESE
    INSTRUCTIONS
540 FOR J=1 TO 100: NEXTJ
550 PMODE 1,1:SCREEN 1,1
560 Q=PEEK(Z)
570 POKE Z,255:POKE Z,0:
    K$=INKEY$:IF K$=""THEN 570
580 POKE Z,Q
590 IF W=0 THEN 610
600 POKE Z,V
610 IF K$="W" THEN W=1 ELSE IF
    K$="P" THEN W=0
620 IF K$=CHR$(10) THEN
    Z=Z+32:GO TO 560
630 IF K$=CHR$(94) THEN
    Z=Z-32:GO TO 560
640 IF K$=CHR$(8) THEN Z=Z-1:
    GO TO 560
650 IF K$=CHR$(9) THEN Z=Z+1:
    GO TO 560
660 IF K$=CHR$(13) THEN
    A=INT(Z/32):Z=32*A
670 IF K$="R" THEN 310
680 IF K$="V" THEN 430
690 IF K$="C" THEN 710
700 GO TO 440
710 FOR J=1535 TO 3070
720 POKE J,0: NEXT J
```

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

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

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

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 Color News
P. O. Box 896
Hartselle, AL 35640

"EARS" BRINGS VOICE RECOGNITION TO THE COLOR COMPUTER

Speech Systems is proud to announce the introduction of EARS (Electronic Audio Recognition System), the first hardware voice recognition device available for the Color Computer. "EARS" is trained by the user and is capable of recognizing any word, phrase, or sound. Up to 64 voice prints can be loaded into memory at one time and subsequently saved to tape or disk. This gives the user the opportunity to assemble an unlimited library of phrases. Programming "EARS" is easy due to the addition of several commands to BASIC such as "Listen", "Match", and others. The cost of EARS is $99.95 + $3 shipping. For more information contact Speech Systems, 38W 255 Deerpath Rd., Batavia, Illinois 60510. (312) 879-6880.

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

QUESTIONS & ANSWERS

QUESTION: Can information be transferred from one computer to another in the same room using your "DYTERM" terminal program?

ANSWER: Our terminal program or any other terminal program can be used for this purpose. It is more straight forward to use the cassette built in software with an audio amplifier to boost the signals. However we don't have a plug in amplifier designed for this purpose and don't know of anyone else who has one on the market.

OPERATING HINT

Use your word processor for writing basic programs. You have the full screen editing capability of the word processor. Save the text as an ASCII BASIC file which can be loaded into your computer as a BASIC program. You can also edit a BASIC program by saving it in ASCII format and loading it into your word processor.

COLOR COMPUTER SOFTWARE

Our software is supplied on Tape and may be saved to a Disk. This saves you money because a disk version is not required.

TERMINAL PROGRAM

DYTERM - 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. $14.95

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. $19.95

MULTIPROGRAM MANAGER (MPM)

The MPM allows up to 5 programs to be loaded into a 32K computer. Run, Delete, or Add programs to the menu. Quickly jump from one PGM to another. $14.95

COMPUTER RESETTER (DYSET)

New Product

A plug in cartridge that allows a CC to be reset when the normal reset fails. Push button switch on cartridge allows the reset function. DYSET cartridge $19.95

ME-5 8-64K Chips	$24.95
6883 SAM Chip	$29.95
SAM Amplifie- module mounts under SAM chip protecting it from shorts	$19.95
6847 VDG Chip	$19.95
6821 PIA Chip	5.95
6822 PIA Chip	6.95
2764 8K BYTE EPROM	9.95
27128 16K BYTE EPROM	14.95

WORD PROCESSING

TELEWRITER-64 -Three Hi-Res screens, true lower case characters, right justification, full screen editor, + menu-driven I/O & format. Tape $49.95, Disk $59.95

WIZ- New Characters set for TW-64 with true descenders and end of line markers Tape $16.95, Disk $21.95

MASTER DESIGN - A text designer/editor to genreate graphics mode lettering with multiple font sizes, textures, shadowing and thicknessess, plus special patterns for creative backgrounds. Screen print routine and Letter Head Utility interface with Telewriter 64 and Basic. Disk $29.95

CARTRIDGE PROGRAMS

We can put your Basic programs in a cartridge and you can have all of your computer's RAM for data. Up to 16K of programs for $39.95.

24 hr phone. Checks, VISA & MC cards. Add $2 ship.

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

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