APRIL 1985
Vol. 2 No. 3

$1.95

DYNAMIC COLOR NEWS

*LOAN INTEREST PROGRAM

* LARGE MEMORY PROGRAMS (Part 3)
* WRITING PROGRAMS (Part 2)
* BASIC PROGRAMMING
* COMPUTER GRAPHICS (Part 3)
* 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 *
* *
* April 1985 *
* *
* Editor and Publisher *
* Bill Chapple *
* *
* Secretary *
* Deanne Hill *
* *
****

CONTENTS

EDITOR'S COMMENTS

The possibility of a 256K upgrade for color computers is very promising. This month we made a 256K prototype but had some design problems so it will not be ready for a while. The prices of 256K chips has dropped and they are in 16 pin packages the same as the 4164 chips used in 64K color computers. The problem is that the SAM chip which refreshes the dynamic RAM chips is designed for a maximum of 64K or a 16 bit address bus. If you are addressing 256K bytes then you will have 18 address lines. So you have to use some kind of design trick to take care of the 2 extra address lines for the 256K chips. To complicate the issue further, the refreshing for the memory chips is accomplished by the SAM which alternately supplies 8 rows and 8 columns to the 4164 chips. For 256K chips we need 9 rows and 9 columns which makes it hard to design for a plug in type upgrade.

We purchased one of the Korean extended basic color computers that Radio Shack is selling for $119.95 on sale. I believe that this is their new price although they advertise it as a sale price. When I looked inside I noticed that it is quite different from the old color 2 computers. There was only one ROM which means that they combined the basic and extended basic ROMS into one package. There were only 2 memory chips which means that they were using 64K memory chips organized as 16K x 4. The 2 gives 16K x 8 which is required for 16K bytes. There was a pair of sockets near the chips which was probably for a plug in upgrade. We want to look at this computer since it represents what they are selling now and an advanced design. The engineers that designed the color computers were very good. So we want to see what they did and look at upgrades for this new computer.

The new computers have most of the chips soldered in. This makes it impossible to upgrade without soldering as most of our inboard upgrades are solderless. As I look back over the various versions of color computers I believe that the best version was the 285 board. I seem to like it the best because almost everything had sockets and it had all of the voltages to the output connector.

In this issue we are continuing with our Graphics series and our Large Memory Program series. Also we are including a finance program that will let you quickly figure the monthly payments for a loan. This should be useful for everyone since we all have to borry money.

We are encouraged with the number of renewals that we have had. Check the number by your label and see if your subscription is expiring. We have received a few letters but would like to hear from more of you. So take a few minutes and write us a letter and let us know what you would like for us to cover in future newsletters.

LARGE MEMORY PROGRAMS
Part 3

We gave machine language subroutines last month that allow the second 32K bank in a 64K computer to be initialized for basic. This was accomplished by transferrring each byte in the first bank to the corresponding byte in the second bank. We also gave a machine language subroutine that allowed exchanging the information in the two banks.

This month we want to show how to use the subroutines. Did

you know that you can write a basic program to start in the first bank and continue in the second bank. This is what we want to look at this month because this will allow longer basic programs to be written. One thing to remember is that when the banks are swapped or exchanged, the variables associated with the old bank remain with that bank. In other words when you exchange banks you exchange variables and all information concerning the banks. This doesn't present any problems though because if both banks are in the "RUN" mode the program will continue.

Transferring Variables Between Banks

What if it is desireable to transfer a variable from one bank to the other. How can this be done? To do this the variable has to be stored in something that is common to both banks. The microprocessor's registers, the PIA registers, and RAM are common to both banks. We will concentrate on this later but as an example if you store a variable in a PIA register using the POKE command and switch banks, then you can recover the variable in the PIA register by using the PEEK command from basic while in the second bank.

If you have a machine language subroutine then you can load a register such as the "B" register with the variable and use the "X" and "Y" registers as pointers. The bank switching is accomplished by storing a value in 65502 or 65503. We showed how to do this last month. In a future newsletter we will show how to move data from one bank to the other.

2 Bank Programs

Suppose you have a large program that is too large for the standard 32K memory bank. There are two methods of partitioning the memory in Color Computers. These are called map type "0" and map type "1". Map type "0" is the standard power up configuration. This allows the lower 32K to be RAM and the upper 32K to be ROM. The ROM area is occupied by the Basic and Extended Basic ROMS plus the cartridge port. There has been much discussion about the disadvantage of this configuration. However if you compare it with other computers you can see that it is a definite advantage. For example with two banks of 32K and a 32K ROM designated memory area there is a total of 96K of memory that the SAM chip can accomodate.

Most other computers do not have this advantage. They operate similar to map type "1" which is all RAM. The ROM information is transferred to RAM which can be done with the Color Computers. This can give about 40K of usable RAM. But with the bank switching scheme there are 64K of usable RAM which is quite an advantage.

Let's say we have a very large program that we want to use in both banks. The simplest way to do this is to divide the program into two relatively equal parts. Put one part in one bank and the other in the other bank. If it is a menu oriented program with more than one menu, then put the program that goes with the first menu in the first bank and the program that goes with the second menu in the second bank. Let one of the options of the menu be to select the other bank.

2 Bank Program Implementation

First load the machine language subroutines we presented last month. Then EXEC 4015 and EXEC 4070. EXEC 4044 exchanges banks. Your program should

start above 4100. For extended basic you can do a PCLEAR 2 or you can do the following: POKE 25,20: POKE 20*256,0: NEW. Remember that EXEC 4070 copies bank 0 into bank 1. You can use this at anytime to make a duplicate of bank 0.

Next enter half of the program in bank 0. When this is run there should be an option in the menu to go to the next menu. When this option is selected then the program should EXEC 4044 to exchange banks.

Go to the other bank by EXEC 4044 and enter the other half of the program. Also put in the menu an option that will allow return to the first bank by EXEC 4044.

Running the Program

After both halves have been entered then type "RUN <ENTER>". Your program should run and display the menu. Select the option that allows you to go to the other bank. Then again type "RUN <ENTER>" and the program should be continuous with access to both banks.

2 Bank Demonstration Program

The following program will demonstrate going from one bank to the other. It just prints information on the screen but you can see that bank switching is occurring. First initialize the computer for 2 bank operation with the machine language subroutines at 4015 and 4070 as previously discussed. Then type in the following program and enter "RUN" from each bank.

```
10 ?"2-BANK DEMONSTRATION PGM
20 ?"BY DYNAMIC ELECTRONICS INC
30 ?"XXXXXXXXXXXXXXXXXXXXXXXXX
40 ?
50 ?"THIS IS THE FIRST BANK
60 FOR J=1 TO 500: NEXT J
70 EXEC 4044: RUN
```

Enter the program in the first bank and EXEC 4070 to copy the program into the second bank. Change "FIRST" in statement 50 to "SECOND" in the second bank. Run the program and the information will be printed on the screen. When the banks switch again enter "RUN" and the program should be continuous. Statement 60 slows down the operation so that you can see what is happening.

COMPUTER GRAPHICS
(Part 3)

Last month we discussed the memory requirements for different graphic resolutions. The graphic resolution can vary from 64H x 32V to 256H x 192V. We can also vary the color for each graphic element. For the lower resolutions we have the full range of colors. As the resolution increases the color options decrease. The highest resolution graphics have only one or two colors.

Also last month we also discussed the cursor and gave a demonstration program that allowed the cursor to be moved across the screen. With the program various patterns could be drawn because a "0" was printed in each location that the cursor occupied. This was an example in using standard characters for drawing images. This is what we do with our printer for generating "DYNAMIC COLOR NEWS" on the front of this newsletter. However this is not what we consider graphics but is just an application using alphanumeric characters.

SEMIGRAPHICS 4

This month we will devote our attention to the semigraphics 4 mode. This mode uses the same display memory as the computer

for normal operation. The display memory location is from 1024 to 1535. The top left element on the screen is location 1024 and the bottom right element is 1535. The format is 32 horizontal characters with 16 vertical lines. This gives a total of 512 characters or bytes. In the semigraphics 4 mode each location is divided into 4 graphic elements. Therefore we can have a total of 4 * 512 or 2048 graphic elements for Semigraphics 4.

Let's look at the format for using semigraphics 4. A byte is composed of 8 bits. The least 4 significant bits determine if the corresponding graphic element is "ON" or "OFF". A "1" is stored if the element is to be "ON" and a "0" is stored if it is to be "OFF". The next 3 elements are used to designate a color value from 0 - 7. The most significant bit is always a "1".

For alphanumeric characters the pixels or picture elements are arranged so that there are 8H x 12V pixels for each character. When this is divided for Graphics 4 there are 4H x 6V pixels for each graphics element. The graphic elements are arranged as follows:

```
**************
*      *      *
*      *      *
*  D   *   C  *
*      *      *
*      *      *
*      *      *
**************
*      *      *
*      *      *
*  B   *   A  *
*      *      *
*      *      *
*      *      *
**************
```

Remember that this is the same area that one character occupies in the normal power up display. If we represent the 3 bits that determine the color as C0, C1, and C2 then the Graphics byte can be composed as follows:

Bits							
7	6	5	4	3	2	1	0
1	C2	C1	C0	D	C	B	A

Now if we let E be the value of the graphic elements A, B, C, and D then

E = 8*D + 4*C + 2*B + A

If we let CL represent the color then

CL = 64*C2 + 32*C1 + 16 *C0
or
CL = 16 * (4*C2 + 2*C1 +C0)

Let's designate the value of the byte as X. Then

X= 128 + CL + E

X is the value we will poke into the display's memory location.

Initialization for Semigraphics 4

It is necessary to do the following pokes for initializing the VDG and SAM chips.

1. A = PEEK (65314): POKE 65314, 128 OR (A AND 7)

2. POKE 65472,0: POKE 65474,0: POKE 65476,0

After doing the pokes we can calculate the value of the byte and store this value in the appropriate memory locations (1024 - 1535).

Semigraphics 4 Demonstration Program

To demonstrate the use of Semigraphics 4 we wrote an example program. The program allows setting any of the 4 graphic areas either "ON" or

"OFF" by entering 4 values which are either "0" or "1". Also the program allows the setting of any of the 8 colors (values 0-7). The 4 arrow keys allow moving the cursor left, right, up, or down one position. When the "R" key is pressed the previous values for the byte are stored in the memory location designated by the cursor's position. If you want to change the color press the "C" key and enter the new color value.

Instructions are included with the program. After the instruction part the screen is cleared and you can then enter the 4 values for A,B,C, and D. The graphic pattern will be displayed in the first location. You can press the "R" key and the pattern will be repeated. When you press the "C" key you can enter a new color value. After changing colors, the values for the graphic pattern should be entered again. You can move anywhere on the screen and draw various patterns. To change a pattern just draw over it.

```
10 PRINT"GRAPHICS DEMONSTRATION
   PGM
20 PRINT"FOR SEMIGRAPHIC 4 MODE
30 PRINT"COPYRITE (c) 1985
40 PRINT"DYNAMIC ELECTRONICS
   INC.
50 PRINT"EACH MEMORY LOCATION
   HAS 4
60 PRINT"ELEMENTS. YOU CAN
   ENTER A 1 IF
70 PRINT"YOU WANT THE ELEMENT
   ON OR A 0  IF YOU WANT THE
   ELEMENT OFF.
80 PRINT"IT TAKES 4 ENTRIES FOR
   EACH
90 PRINT"BYTE. THE ORDER IS AS
   FOLLOWS
100 PRINT"          D C
110 PRINT"          B A
120 PRINT"YOU ENTER VALUES IN
    THE ORDER   A,B,C,D. AFTER
    ENTERING THE 4   VALUES THE
    RESULT WILL BE
130 PRINT"DISPLAYED. PRESS
    ENTER KEY.";
140 INPUTR:PRINT
150 PRINT"THE 4 ARROWS MOVE THE
    CURSOR IN THE INDICATED
    DIRECTION. YOU CANCHANGE
    THE COLOR BY PRESSING <C>AND
    ENTERING A VALUE FROM 0-7
160 PRINT"FOR A NEW COLOR.  YOU
    CAN RE-   PEAT THE LAST
    VALUES BY PRESSINGTHE <R>
    KEY. THE <ENTER> KEY
170 PRINT"RETURNS TO THE LEFT
    ELEMENT
180 PRINT"PRESS ENTER
    KEY":INPUT Q
190 PRINT:PRINT"AFTER ENTERING
    THE FIRST 4
200 PRINT"VALUES YOU CAN START
    DRAWING ON THE SCREEN.
210 PRINT"PRESS ENTER
    KEY":INPUT Q
220 N=4:CLS:GO SUB 480
230 M=1024: A=PEEK(65314):
    POKE65314,128 OR (A AND 7)
240 POKE65472,0: POKE65474,0:
    POKE65476,0
250 V=0:FOR J=1 TO N: GO
    SUB320: X(J)=X:IF X(J)>0
    THEN X(J)=1
260 K=2HJ-1):V=V+X(J)*K
270 NEXT J:X=V: IF X>63 THEN
    250
280 GO SUB 310
290 W=128+16*C+X
300 POKE M,W:M=M+1:GO TO 250
310 PRINT@448,X;:FOR J=1 TO
    200: NEXT J: RETURN
320 GOSUB 430: X$=INKEY$: IF X$=""
    THEN 320
330 POKE M,223:IF X$="B THEN
    M=M-32:GO TO 320
340 IF X$=CHR$(10) THEN
    M=M+32:GO TO 320
350 IF X$=CHR$(8) THEN M=M-1:
    GO TO 320
360 IF X$=CHR$(9) THEN M=M+1:
    GO TO 320
370 IF X$="C" THEN
    PRINT"COLOR="C;:INPUT"COLOR";C
380 IF C>7 THEN 370 ELSE IF
    X$="C" THEN 320
390 IF X$="R" THEN 300
400 IF X$=CHR$(13) THEN M=INT
    (M/32)*32:GO TO 320
410 X=ASC(X$)-48:RETURN
420 RETURN
430 Z=Z+1:IF Z<223 THEN Z=223
```

```
440 IF Z>224 THEN Z=223
450 POKE M,Z
460 RETURN
470 'PUT 0 IN EACH MEMORY
     LOCATION
480 FOR J=1024 TO 1535: POKE
    J,223:NEXT J:RETURN
```

```
+ ++ + ++ + ++ ++ ++ + ++ + ++ +
+                               +
+         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 Color Computers. This +
+ is the last issue for those +
+ with 4/85.                   +
+                               +
+ ++ + ++ + ++ ++ ++ + ++ + ++ +
```

Writing Programs

Part 2

Last month we introduced Basic and talked about program structure, the NEW command, REMARKS, LIST and LLIST. This month we want to continue and give you some techniques so that you can begin to write programs. As a review we use statements to contain Basic commands. These statements are numbered from 0 to over 60000. Basic starts with the first number in the program, does the command in that statement and goes to the next numerical statement unless directed to go elsewhere. You can delete statements you don't want by typing the statement number and pressing <ENTER>. If you want to replace a statement with a new one then you can type in the new statement and press <ENTER>. The old statement will be replaced with the new one.

Last month we talked about REMARK statements. These are used when you need to put information or labels within a program. This is for your own benefit and Basic ignores the information. We recommend using remarks whenever you write a program so you can keep things straight. If you have a simple program that does only one task then it is not hard to remember what is happening. However if you have a long program with many subsections then remarks will help you to identify the purpose of each section.

Variables

A variable is an alphanumeric character or characters that stand for an unknown. The term "alphanumeric" means the letters of the alphabet, punctuation, and the numbers. All of the characters and symbols on the keyboard are alphanumeric. There are two types of variables which are numerical and string.

Numerical Variables

These are the variables that are used in an equation. They can be composed of one or more alphanumeric characters. The first character has to be a member of the alphabet and the second character can be a number or a member of the alphabet. Examples of numerical variables are:

X, A, ZA, Y1, FK, GR

String Variables

One of the main advantages of Basic is its ability to handle word and word phrases. Computers were first designed to handle numerical calculations. Now with the enormous amount of printed material that is required, it is advantageous for computers to also handle words. Newspapers, magazines, publishers, attorneys, plus many other businesses have tremendous

requirements for writing letters and text. Basic can be used for word processing for smaller writing requirements. For the large writing needs a dedicated word processor should be used.

A string variable stands for a group of alphanumeric characters. These variables are identified by the "$" sign at the end of the variable. Examples of string variables are:

X$, Z2$, AA$, P$, Q5$

Entering Variables

Variables can be entered directly from the keyboard or defined within basic statements. Let's look at defining variables within basic statements. A basic program segment follows with numerical and string variables.

```
10 'X IS THE VALUE OF THE CHECK
20 X=25
30 'X$ IS THE PERSONS NAME
40 X$="JAMES SMITH"
```

Notice that we used remarks before each variable to identify them. In statement 20, X is defined to equal 25. In statement 40 X$ is defined to equall "JAMES SMITH". Notice that parenthesis are at the beginning and ending of string variables.

The Print Command

Print is one of the most used commands. Print means that information is going to be displayed on the television screen. Since the television screen is the device that provides communication from the computer to you, it is important to know how to effectively use the "PRINT" command. You communicate with the computer by the keyboard and the computer communicates to you by the television screen. The print command has two forms. The simple form is the "?" mark. The long form is "PRINT". We recommend using "?" because it is shorter and easier to type. When a program is listed the word "PRINT" is displayed instead of the question mark.

Type in the previous program where we defined X and X$. Then type "RUN" <ENTER>. We will be using the notation <ENTER> to mean press the "ENTER" Key. The computer will give an OK. Now from the keyboard type ?X;X$ <ENTER>. Notice that 25 JAMES SMITH is printed on the screen. Both variables are printed side by side. The ";" between the variables means to print the second variable beside the first one.

Now type ?X,X$ <ENTER>. What did you notice? The "," between the variables means to print the second variable in the next field. A field is 16 spaces and since there are 32 spaces across the screen the first variable is printed starting at the left. The second variable is printed starting at the center of the screen.

PRINT @ or ? @

The PRINT @ command allows a variable to be printed at any location on the screen. There are 32 spaces across the screen and 16 lines down. This gives a total of 512 locations. The following is for using PRINT @.

```
? @ 320,X <ENTER>
```

If you will type in the preceeding statement after running our test program, then the value of X will be printed starting at location number 320. This is 10 lines down from the top of the screen. Locations are determined by designating the upper left location as "0". The upper right location is "31". The left position for the second line is "32" and the bottom right position is "511". Therefore the format for using ?@ is

as follows:

```
? @ L, X or PRINT @ L,X
```

where L is the number ofloc-ations to skip and X is the variable to be printed.

Now enter the following:

```
? @ 64,,X$ <ENTER>
```

The characters representing X$ should be printed starting at the middle of the third line. The extra comma forces the printing to move to the next field as discussed earlier.

Printing to a Printer

To print to a printer use the same format as printing to the screen except insert "#-2,". Suppose we want to print the variables X and X$ to the printer. Then we would enter the following print command.

```
?#-2, X, X$
```

X will be printed in the first column on the printer and X$ will start printing at the second field or column number 16. The "," and ";" force the same printing formats for the printer as they do for printing on the screen. The ";" causes variables to be printed beside each other.

STRING PRINTING

A string which can be in-structions can be printed without assigning a variable name to them. The procedure is to enter

```
?"XXXXXXX XXX" <ENTER>
```

The characters between the parenthesis will be printed. This can be entered from the keyboard or incorporated within a program.

CLEAR SCREEN Command

Sometimes before printing re-sults it is desireable to clear or erase the screen. This can be done from the keyboard by pressing the "CLEAR" key. It can also be incorporated into Basic Programs by entering CLS as a command.

Print Demonstration Program

The following program will put a title box around a name.

```
10' PRINT DEMONSTRATION PROGRAM
15' CLEAR THE SCREEN FIRST
20 CLS
25 X$="*************************"
30 Y$="*                        *"
35 Z$="*    JAMES SMITH         *"
40 ?X$
45 ?Y$
50 ?Z$
55 ?Y$
60 ?X$
```

Type in the program and run it by typing "RUN" <ENTER>. The following should appear on your screen.

```
*************************
*                       *
*    JAMES SMITH        *
*                       *
*************************
```

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

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

Computer Resetter (DYSET)

Dynamic Electronics Inc. is pleased to announce a plug in module that will allow color computers to be reset when the normal reset fails. The name of the product is "DYSET". You are working on that special program and suddenly the computer hangs up. The normal reset will not work. However DYSET is plugged in and pressing the push button switch on the cartridge causes the computer to reset. The cost of DYSET is only $19.95 + $2 shipping. For more information contact Dynamic Electronics Inc., Box 896, Hartselle, AL 35640. (205) 773-2758.

LOAN INTEREST PROGRAM

At one time or another we must all sign up for a loan to purchase the things we want. Large loans for homes and automobiles are considered standard practice for most people. However before you sign on the contract wouldn't it be nice if you could look at the variables such as length of time, interest rate and payments. The following menu oriented program will allow you to do just that. You can vary one variable at a time and the results will be displayed. For example what is the difference in payments for $30000 over a 20 year period and a 30 year period? The program allows changing just the time and the new payments will be displayed. The program is easy to run and we hope you will enjoy using it.

Loan Interest
Program Listing

```
10 ?"INTEREST PROGRAM WRITTEN
14 ?"COPYRIGHT (c) 1985
16 ?"DYNAMIC ELECTRONICS INC.
20 N=12'PAYMENTS PER YEAR
30 M=12
35 'PRINT THE VARIABLES
40 ?"INTEREST PROGRAM
50 ?"1"M"MONTHS FOR LOAN
60 ?"2"Y1"YEARS FOR LOAN
70 ?"3 $"P"PRINCIPAL
80 ?"4"I"ANNUAL INTEREST RATE
90 Y=M/12 + Y1:A=(.01*P*I)/N:
   B=(.01*I)/N
100 C=(B+1)↑(N*Y):D=(C-1)/C:
```

```
    R=A/D
110 R=INT(R*100+.05)/100
120 ?"5 $"R"MONTHLY PAYMENTS
130 ?"PRESS P TO PRINT
    RESULTS
135 '140 WAITS FOR A KEY
140 X$=INKEY$:IF X$="" THEN 140
150 IF X$="1" THEN INPUT
    "MONTHS";M: GO TO 40
160 IF X$="2" THEN INPUT
    "YEARS";Y1:GO TO 40
170 IF X$="3"THEN INPUT
    "PRINCIPAL";P:GO TO 40
180 IF X$="4" THEN INPUT
    "INTEREST";I:GO TO 40
190 IF X$="P" THEN 200 ELSE 40
195 'PRINTER PART
200 ?#-2,Y1"YEARS";M"MONTHS",
210 ?#-2,"$"P"PRINCIPAL"
220 ?#-2,I"% ANNUAL INTEREST
     RATE",
230 ?#-2,"$"R"MONTHLY PAYMENTS
240 ?#-2,CHR$(10)
250 GO TO 40
```

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

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

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

****

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

Call anytime (205) 773-2758

DYNAMIC ELECTRONICS INC.

P. O. Box 896

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