ENGINEERING NOTES
on
Radio Shack Color Computers

August 1984 $1.95
Vol. 1 No. 6

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


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


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

* CHECK BOOK PROGRAM
* WORD PROCESSORS
* POWERFUL REMARKS
(Part 2)

* COMPUTER THEORY
* BASIC PROGRAMMING

* MACHINE LANGUAGE PROGR.
* LETTERS with COMMENTS

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.

All paid subscribers are entitled to discounts of 10% on hardware. and 20% on software manufactured or produced by Dynamic Electronics Inc. plus "specials" mentioned in the newsletter. To receive these discounts use your DCN number which is at the right of your name on the address label. DCN subscribers may obtain a personalized reply to specific computer problems or advice on purchasing equipment. The charge for this service is $10.

DYNAMIC COLOR NEWS

August 1984

Editor and Publisher
Bill Chapple

Secretary
Helen Earls

CONTENTS

EDITOR'S COMMENTS

July was a very good month for us. I want to thank each of you who took advantage of our Summer Specials. We have received some letters about our products and this newsletter and certainly do appreciate hearing from you. Generally it looks like our customers are satisfied with our products and newsletter. A couple of customers had problems with the 96KX module and could not get them to work. This is to be expected with hardware modules where inexperienced people are installing them. Although they are designed to be as easy to install as possible, sometimes an individual can't get his to work. This represents only a small portion of the people who have purchased the modules and we are available to install them in your computers if you send them to us. We had a couple of customers who sent us their computers for installing our upgrades last month.

This newsletter is not intended to be a sales scheme but is to provide technical information as stated on page 2. However I think that comments about our products are in line because you can have a better understanding of what they will do if we discuss them. There is quite a misunderstanding about which computer accessories are required for a particular job and after reading our comments you should at least know what ours will do and maybe understand enough to purchase a competitor's product. For example suppose you want to add a printer to your computer. We covered the use of ASCII and terms pertaining to printers such as BAUD RATE. So you should have an understanding of what is required to hook up a printer or MODEM.

A check book program is given in our Powerful Remarks section. This is how we keep our records and the program is similar to the one we use. Also in the machine language section we show how to find the memory location where a remark statement is located. Actually this works for any statement and not only for remark statements. Also this month we have an editorial on word processors. They can do much more than just write editorials. So you will want to read that editorial if you have a word processor or are considering purchasing one.

We have finalized our design on our uninterupted power source (UPS). It consits of a switch to enable it, a light emitting diode (LED) to show when you have power to the RAMS or 5 volt buss, and a control circuit. The battery can mount under the keyboard of color computers and the UPS is usable with all computers that have 5 volt RAMS. The control circuit mounts on the battery and one wire has to be soldered to the unregulated 5V source (about 9 volts). Two wires with eyelets are supplied to mount on leads of an integrated circuit. If you are tired of loosing your programs when power fails then you need a UPS.

As stated earlier we have received some letters and appreciate those who wrote. Let us know what you think as your input helps guide us in designing new products and improving our old ones. - Bill.

POWERFUL REMARKS
PART 2

Last month we showed how to use remark statements to contain information for word strings. We also showed how a simple word processor could be constructed with each sentence being contained within a remark statement number. The sentences can be edited by retyping them or by using the extended basic editor. This month we want to show how to use remark statements to store data. Information in Data statements has to be perfectly ordered and read in sequence. This slows the computer down and puts a hardship on the programmer because he has to keep a record of what has been read. Our approach here will be to go to statement number X and get the information contained in that statement for our data.

Let's give an example of the usefulness of this approach. We have a program for our check book and have reserved remark numbers 1000-1999 for the first month, 2000-2999 for the second month, etc. So the beginning of data for each month is the number of the month times 1000. If we want to review the checks for the sixth month, we do not have to know anything about the first 5 months. Our program searchs through the statement numbers until it finds statement number 6000 and then starts displaying our checks, expenditures, deposits, and balances after each transaction.

DATA in REMARKS

To develop a data processing program we need to decide on what we want in the remark statements. There are two kinds of data, string and numerical. It would be nice if we could enter information similar to the way it is entered in DATA statements. The use of a comma to seperate data elements is a disadvantage especially if you want to enter addresses where the City and State are seperated by a comma. So let's use a colon instead of the comma to seperate the data. We are familar with the use of the colon for stacking basic commands within one statement number so this should be an easy adjustment to make. Now let's decide how we want the data to be arranged and what will be our plan of attack?

Last month we removed a byte from memory with the PEEK command, converted it to a string with the CHR$ command, and printed it to the screen or an external printer. This is fine for printing words or character strings. However for handling data the process is a little more complicated. The following is a list of things that need to be done after we find the beginning of our data in the remark statement.

1. Find the memory for the statement number and the first byte of data. We did this last month. We use A$ and B$ as our working string variables. So they need to be emptied by the following commands before we look at the first byte. A$ = "" : B$ = "".
2. Get the next byte by letting A = PEEK (M) and convert this to a string (A$). A$ = CHR$ (A).
3. Check to see if A$ is equal to a colon or if A is equal to zero? If so we have enough information for our data and should go to step 5.
4. B$ = B$ + A$. Combine the last character with the previous character string. Increment M and go to step 2.
5. B$ contains the string data. Let the variable you are looking for be equal to B$. For example suppose you are looking for X$ Then write a statement that equates X$ with B$. If the variable is numeric, B$ can be converted to a numerical variable by using VAL (B$). So you can put in your program X = VAL (B$) to find the numerical value of the string B$.

Following is a listing of the check book program with data in statements 1000-1999 and 2000-2999. Notice in a statement near the beginning of the data block we have a statement establishing GT. This is the beginning balance at the start of the month.

```
10 80 TO 60000
1000 'THIS IS THE FIRST MONTH
1005 GT=-448.05:RETURN
1010 '128:1-2:C:JO ANN:7.87
1020 '129:1-2:C:ALA STATE REV:14.06
```

```
1030 '01:1-3:D:SALES:515.85
1060 '02:1-5:D:SALES:200.7
1070 '132:1-5:C:POST OFFICE:29.1
1080 '133:1-7:C:WAYNE GREEN:208.25
2000 'SECOND MONTH OR FEBRUARY
2002 GT=-311.53
2005 RETURN
2010 '155:2-2:C:POST OFFICE:9.4
2015 '1:2-1:D:SALES:77.85
2020 '156:2-8:C:TIM CHAPPLE:10
2025 '2:2-6:D:SALES:170.9
2030 '157:2-10:C:RADIO SHACK (OS-9):74.85
2035 '158:2-10:C:RADIO SHACK:16.3
2040 '7:2-11:D:SALES:250.95
2070 '4:2-16:D:SALES:44.9
2075 '5:2-17:D:SALES:55
2080 '164:2-16:C:JO ANN:91.43
3000 'MARCH ENTRIES
4000 'APRIL ENTRIES
5000 'MAY ENTRIES
6000 'JUNE ENTRIES
7000 'JULY ENTRIES
8000 'AUGUST ENTRIES
9000 'SEPTEMBER ENTRIES
10000 'OCTOBER ENTRIES
11000 'NOVEMBER ENTRIES
12000 'DECEMBER ENTRIES
59999 '
60000 'THIS IS THE PROGRAM
60010 INPUT"ENTER 1 FOR PRINTER";P
60020 INPUT"NUMBER FOR THE MONTH";N
60030 ON N GO SUB 1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,
      11000,12000
60040 PRINT"BEGINNING BAL="GT
60050 IF P=1 THEN PRINT#-2,"BEGINNING BAL="GT;"FOR MONTH # "N;CHR$(13)
60060 ST=1000*N+10
60070 GO SUB 60180
60080 PRINT"# "CN;"  "DC$;" $"VA
60090 PRINT"TO "TW$;"  ON "DT$
60100 IF T$="C" THEN T=-1 ELSE IF T$="D" THEN T=1
60110 GT=GT+VA*T:GT=INT(100*GT+.5)/100
60120 PRINT"TOTAL="GT
60130 PRINT:IF T$="C"THEN X$="-"ELSE IF T$="D"THENX$="+"
60140 IF P=1 THEN PRINT#-2,CN;"  "DC$,"$";X$;VA,"BAL="GT,TW$;"   "DT$
60150 GO SUB 60370
60160 IF CN=0 THEN 60020
60170 GO TO 60080
60180 'THIS FINDS MEMORY FOR STATEMENT NUMBERS
60190 X=256*PEEK(25)+PEEK(26):Y=X+2
60200 NS=256*PEEK(X)+PEEK(X+1):SN=256*PEEK(Y)+PEEK(Y+1)
60210 IF ST>SN THEN X=NS:Y=X+2:GO TO 60200
60220 M=Y+4:GO TO 60370
60230 '
60360 RETURN
60370 'THIS CALCULATES THE CHECK NUMBER (CN)
60380 GO SUB 60550
60390 CN=VAL(B$)
60400 '
```

```
60410 'THIS DETERMINES THE DATE THE TRANSACTION WAS MADE (DT$)
60420 GO SUB 60550
60430 DT$=B$
60440 'THIS DETERMINES IF CHECK OR DEPOSIT (T$)
60450 GO SUB 60550
60460 T$=B$
60470 '
60480 'THIS DETERMINES TO WHOM THE CHECK WAS WRITTEN (TW$)
60490 GO SUB 60550:TW$=B$
60500 '
60510 'THIS DETERMINES THE AMOUNT OF THE CHECK (VA)
60520 GO SUB 60550
60530 VA=VAL(B$):M=M+7:RETURN
60540 'THIS DETERMINES THE CHARACTERS BETWEEN COLONS
60550 A$="":B$=""
60560 A=PEEK(M):A$=CHR$(A)
60570 IF A$=":" THEN 60590 ELSE IF A=0 THEN RETURN
60580 B$=B$+A$:M=M+1: GO TO 60560
60590 M=M+1:RETURN
```

Rather than put a lot of comments in the program we decided to give the program with some entries for the first and second month and then to explain how the proram works. The program starts at 60000. The GO TO 60000 in statement 10 transfers the program to statement 60000. Notice the entries from 1000 to 1080 for the first month. In 1005 GT is the balance at the beginning of the month. This is a subroutine that is called from the main program. The check book entries start at 1010.

This program supports a printer and a 1 is entered in 60010 for a printer and the number for the month is entered in 60020. Notice statement 60030 calls a subroutine at the beginning of each data block to determine the beginning balance (GT).

The subroutine at 60550 actually removes the data from memory. This is what you need to know if you want to write your own program. The rest of the program is similar to the one we gave last month.

If you have a printer and select a 1 for printer when you run the program then the print out will be as shown below. The columns are (1) check number or deposit number, (2) amount, (3) balance after the check or deposit, (4) to whom the check was written followed by the date.

```
BEGINNING BAL=-448.05 FOR MONTH # 1

128               $-7.87           BAL=-455.92     JO ANN   1-2
129               $-14.06          BAL=-469.98     ALA STATE REV 1-2
1                 $+515.85         BAL=-45.87      SALES    1-3
2                 $+200.7          BAL=-246.57     SALES    1-5
132               $-29.1           BAL=-217.47     POST OFFICE    1-5
133               $-208.25         BAL= 9.22       WAYNE GREEN    1-7


BEGINNING BAL=-311.53 FOR MONTH # 2

155               $9.4             BAL=-320.93       POST OFFICE     2-2
1                 $+77.85          BAL=-243.08       SALES      2-1
156               $-10             BAL=-253.08       TIM CHAPPLE       2-8
2                 $+170.9          BAL=-82.18        RADIO SHACK  (OS-9)  2-10
157               $-74.85          BAL=-157.03       RADIO SHACK  2-10
```

158	$- 16.3	BAL=-173.33	SALES	2-10
7	$+ 250.95	BAL= 77.62	SALES	2-11
4	$$+ 44.9	BAL=-122.52	SALES	2-16
5	$+ 55	BAL=-177.52	SALES	2-17
164	$- 91.43	BAL=-86.09	JO ANN	2-16

WORD PROCESSORS

You may wonder why we would cover a subject such as word processors. If you are involved in any type of business activity you will have paper work to do. The amount of paper work of course varies with the type of business. For our operation we have to write letters, compose advertisements, write new product releases, design a catalog, write computer programs, plus our newsletter. Even if you are not involved in a business activity, a word processor can be of tremendous advantage to you.

We want to discuss some of the things that can be done with word processors. Of course the options vary depending on the particular features of the word processor you are using. Here are some of the things that you can do with word processors.

1. Write letters. You can enter your letterhead, date, and closing signature once and then type in each letter. The parameters for the top, bottom, and left margins can be set once and will be the same for all letters. The letter can be printed once or twice if you want a copy of it for your file. Also the word processor can address your envelopes.

2. Write editorials. We do a lot of editorial writing such as this article. Term papers, reports, brochures, newsletters, and catalogs, are examples of items that fit into this group.

3. Compose and edit Basic Programs. With a word processor you can move to any place within the text. This makes it easy to compose Basic programs because you can make changes at will. If the word processor can save programs in ASCII format then they can be used for composing and editing Basic programs.

4. Inventory. Because word processors generally have the capability of searching for word strings, they can be used to find items. They can be used for inventory or for searching files.

5. Diary. Although similar to item 4, personal or business records of events can be entered. The search feature can be used to find information entered by date or any other word string.

MODIFYING WORD PROCESSORS

Now that we have mentioned a few things word processors can do let's look at what they can't do. If you want it to do something different from what it was designed then it is almost impossible to make a patch to change it unless you are very good at machine language programming. What we are trying to say is that you have almost no options to modify the processor. If it doesn't do something you like or need then forget it.

For example let'e take a disk system with a word processor. For some processors you load in the processor and run or execute it. If you know the name of a file to edit then load it in. Before you can do anything else a new file has to be created with a new name. Then some preliminary information about this file is stored on the disk. After you have finished the file then enter an end of text command and the file will be saved to a disk. For some word processors you can not print your text from the word processor. You have to save your information to a disk and then print a file from the disk to the printer.

You might wonder if there are limitations then what can they do? They can allow you to write sentences easily. You can insert or delete letters to correct mispelled words. Move blocks of sentences from one place to another. Delete or erase blocks of words. Word processors also allow upper, lower, and left margins to be set. Also some will justify or line up the right margins. Some have character generators that generate true lower case characters and allow various numbers of characters per line to be displayed on the screen.

WRITING BASIC PRORAMS

As mentioned earlier we write most of our basic programs initially on a word processor. To do this the word processor has to have the ability to save files in ASCII format. We covered ASCII in previous issues of this newsletter. If you will refer to instructions that came with your computer, you will find that Basic programs can be saved in ASCII format by adding a commma and an A after the program name. For example suppose we have a Basic program named WP-3 and we want to save it to a disk in ASCII format. Then we would type in the following:

```
SAVE "WP-3", A
```

Now if we compose a basic program using a word processor and saved it on the disk as an ASCII file, then when we gave the command to load the program it would be loaded the same as any ASCII basic program.

Before saving a basic program composed on a word processor, it is necessary to make sure that each line starts with a statement number. Use the word processor to go up and down through the program and make sure that each statement number is the beginning of a line. Don't have any blank lines as basic won't know how to use this.

COMPOSING FILES

The powerful search feature of word processors make them very useful for inventory, recipies, geneologies, or music files. Label each file with a character string that will be easy to remember and that is unique so the word processor can find it for you. Then you can enter the information in any order you like and the word processor will find it for you.

Writing Text

The purpose of word processors is to allow text to be easily written. All have a blinking cursor of some sort to let you know where you are within the text. When the processors are in the edit mode, you can insert characters anywhere within the text, delete characters, and copy text from one place to another.

Depending on the word processor, there are 2 or more menues to allow selection of features. One menu is usually devoted to printer parameters such as margines, number of characters per line, number of lines per page, and baud rate. Also printer commands can be embedded within the text.

Summary

If you do a lot of writing you need a good word processor. For $60 you can puchase a very good one. Also it can be used for generating files, geneologies, inventories, and writing Basic programs.

MACHINE LANGUAGE PROGRAMMING

Last month we continued with interrupts. This month we want to show you how to find statement numbers for Basic programs. Because of the powerful organizational capabilities of Basic, we use it for writing our programs and use machine language subroutines when we need to perform a task that is not easily accomplished by Basic or when we need more speed. Also machine language subroutines can be carried within remark statements with Basic programs. This makes things nicer as the machine language subroutines become a part of the program and don't have to be loaded separately.

There are two things that need to be done to carry machine language subroutines within remark statements. (1) First of all we need to find where the machine language subroutine starts. A machine language subroutine can be called from Basic with EXEC M where M is the location of the subroutine within the memory map. We have shown how to find the memory location for a basic statement using Basic. We want to show how to do this with machine language subroutines. (2) The second thing to be aware of is that a zero cannot appear within a remark statement. Basic will interpret this to mean the end of a Basic statement. So the subroutines have to be written so that a zero does not appear within the remark statements.

PROGRAM DEVELOPMENT

To develop any kind of program you have to know what you want the program to do. We want a machine language subroutine that will quickly find the memory location for a given statement number. Let's list what we want our program to accomplish for us.

1. Reserve a memory area for the subroutine to use for vectors. Let's reserve 590 and 591 for the vector of the start of the Basic program. Also let's reserve 592 and 593 for the value of the statement number, and 594 and 595 for the value of the memory where the statement starts.
2. Our program will use the vector in 592 as the statement number we want to find. It will start with the vectors in 590 and 592 and continue until it finds the statement number.
3. The memory vector for the statement number will be stored in locations 594 and 595.

One of the subroutines of our 96KX software finds statement numbers for a Basic Program. It is located at 57677 or $E14D. The program is position independent and can reside anywhere in memory. The program is as follows:

```
57677   LDX E 590  Load the X Register with the Vector in 590
57680   LDY indexed to X with offset=+2 (Statement
Number)
57683   LDD indexed to X with 0 offset (Load D with next statement Vector)
57685   EXG D and X  (Exchange contents of D and X Registers)
57687   CMPYE 592  (Compare Y with statement number)
57691   BLO  57680  (Go to 57680 if Y is lower than Statemnt Number)
57693   EXG  D, X    (Exchange D and X)
57695   LEAX  X, 2  (Load X with X + 2)
57697   STX E  594   (Store X in 594) This is the answer we want
57700   RTS       (Return from the subroutine)
```

The program works very similar to the Basic program we gave last month except the beginning vector is in 590 instead of 25. This allows us to find the memory location of statement numbers for any Basic program which resides within the memory map.

The decimal values for the subroutine are as follows. You can relocate them to some memory location such as 4000 if you wish to use the program.

```
190,  2, 78, 16, 174,2, 236, 132, 30, 1, 16, 188,2, 80, 37, 243, 30, 1, 48,
2,  191, 2, 82, 57
```

To use the program put the beginning of basic vector in 590 & 591 and the statement number in 592 & 593. Then EXEC M where M is the memory location of the start of the subroutine. Your answer will be in 594 & 595 in vector format.

LETTERS

****

Hello Bill,

Your news letter really does fill the void in an easy and educational way. Thanks for sending all my back issues together. I didn't want to miss anything.

You asked for ideas or request from your readers and I thought you might find the enclosed article interersting enough to comment or discuss in one of your columns. I would like to see if it would be possible to connect my video camera (color) to my CoCo. In this way digitizing scenes, portraits , etc. Unfortunatly, I don't understand the article completely to use it.

Can 256 DRAM chips be used in your 128K upgrade board. I know they're $60.00 each, but if digital video on digitized voice systems work best with more memory then it may be cost effective for me.

Best of luck,

Arthur Cortes

ANSWER:

The article Arthur sent showed how to digitize video so pictures could processed by computers and saved. I recently saw an article or new product release of such a device. I wrote for information but haven't received it yet. This is another of the many uses of color computers.

As for the 256K RAMS they cannot be used with our 128K upgrades. The problem is that the microprocessor cannot address more than 64K of memory. The SAM chip allows the upper 32K of memory to be reserved as ROM and the lower 32K can be used for two 32K banks of RAM memory. This gives only 96K. There are memory management chips that are available to allow more memory to be used. We looked at them when we came out with our 128K upgrades and decided not to use them because they would take more space than the modular design we have. The 256K chips can be used of course but they are not plug in replacements for the 64K RAMS. We are looking at a 256K upgrade and may have news about it next month.

****

Dear Sirs:

As a member of your news service (#108) I thought I would drop you a line and let you know what I feel, as a CoCo owner, and what my needs are.

First off, I feel that your service is a good one, even if I haven't been able to understand all of it.

I did purchase your 96KX but since then you came out with the 96KX-M unit. That would be a lot easier to use and I wish to know how much it would cost to update.

For my use of the color computer, which is most of the time as a word processor and large filing system, I would really like to see more RAM than 4 units of 32K. The "Telewriter 64" puts me in 64K but I need about 45K more. What are your plans for upgrading to something like this?

Also there is a unit on the market that I am sure you have heard of that will give us an 80 column display and CPM compatability. Are you headed towards something like this or ?.

Thank you

William Ward

ANSWER:

First of all we will exchange any of our 96KX cartridges with output connectors for our 96KX-M for $10 + $2 shipping. Although the 96KX-M is cheaper than our original cartridge, and since we are no longer making this item we can only scrap it for parts.

As for the additional memory we have our 128K expanders but need to make patches for the Telewriter to use it.

The 80 column display requires a video monitor.

The reason you can't get 80 columns on a standard television is because the color computers are designed using the Motorola MC 6847 video display chip. The reviews I have read are favorable for the 80 column display but you need to also read the reviews and literature to make sure this is what you want.

As for the CP/M adapter I think this would be worthwhile if you have another computer that has CP/M. I have seen two advertised for $250 and $300. This seems reasonable if you want this feature.

At present we have no plans to enter these areas. We will probably come out with 256K memory expanders in the near future and continue along these lines unless response from our customers and readers changes our mind.

****

Write us a letter if you have a question or an operating hint. We will print your letter with answers here. These are helpful to us in that they show us what you need.

Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I understand I will receive a monthly news letter, Discounts on DYNAMIC ELECTRONIC INC. Computer products 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 ____

Address __________________________________

City ______________________________________

State & Zip ______________________________

Enclosed is a check ___

charge to VISA ___ MC ___ Number ________________________Exp._____

Mail payment to
Dynamic Electronics Inc
P. O. Box 896
Hartselle, AL 35640

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