

# SYNTAX

2.4

SEPT - OCT 1986

## TABLE OF CONTENTS

Letters.....	2,3,4,5,6
Adam News and Developments.....	7
Adam Graphics: The Forty Column Text Mode.....	8,9
Machine Language Primer.....	10,11
E-Z EDIT.....	12,13
Hardware Review: Adam Disk Drive.....	14
Program Reviews: Expertype.....	15
Smart Letters & Forms.....	16
Hard Hat Mack/Pinball Construction.....	17
Programs.....	18,19
FCAUG Product List.....	20



FIRST CANADIAN ADAM USERS' GROUP  
P.O. Box 547 Victoria Station  
Westmount, P. Q.  
H3Z 2Y6

POSTAGE  
PAID

Mt1. PQ  
Permit  
NO. 7152

L E T T E R S

Dear Sir,

Being a new member, I have just finished reading through all the SYNTAX back issues. I am delighted with the quantity and quality of the information provided. Keep up the good work.

I have contacted Joanne Allen about acquiring Adam's Companion but it would seem to be out of print. If any member finds or has a copy to sell or loan, I would be most appreciative. Ms. Allen was prompt and courteous and supplied a list of alternate books. In the event that I am not able to get Adam's Companion, could you suggest an alternative book which will expand on graphics, sound, and machine language routines.

Guy-R. La Forest  
St. Jacques, N.B.

Adam's Companion was one of the first and better books written on the Adam. Since many copies have been sold, it should be possible to buy it second hand. We will try to locate a copy for you and let you know of the cost. The Basic Adam appears to be a good alternative choice according to fellow member Mike McDermott. It is described as, "a self-teaching guide that assumes the reader has no prior computer experience. It comes complete with programs that are sophisticated enough for those who have already acquired good working skills".

Dear FCAUG,

One book for the Adam that I found useful is The Basic Adam by William Abikoff and Gary Cornell, (Wiley Press \$22.95). As a computer professional, I found all the other Adam books to be oriented more to the complete novice, and mostly an attempt to do what Coleco's programming manual should have done but didn't. The Basic Adam devotes fewer pages to introducing the computer and setting it up, and is more ambitious in its examples. I find it makes a good addition to Adam's Companion.

Mike McDermott  
Willowdale, Ont.

Dear FCAUG,

I have applied your technique as described in the July - August 1985 issue for "Printer Ribbon Surgery". It is an excellent device which saves lots of time and money. I would like to inquire if someone has tried to interface the Adam to another printer which has characteristics similar to SmartWriter. I would also like to know why the cursor stops and doesn't move to any position while working with the word processor and sometimes the words in the text change to different characters. Is it because I use the computer for more than five or six hours at a time?

S. Vashishta  
Edmonton, Alta.

As discussed in our last issue, it is now possible to use other

printers with the Adam using the PIA2 centronics interface. Since you use the computer mainly for word processing, you might want to consider another letter-quality printer. As for your problem with cursor freeze ups, our own experience has shown that this is due to overheating although some Adams are more prone to it than others. Make sure the air vents of all your components are not covered. Provide a minimum of three inches of vertical clearance and six inches of horizontal clearance for each component from the back of any desk or wall. This is especially critical for the printer as it generates considerable heat. It is also a wise practice to shut off your system every 4 or 5 hours after using it, to let it cool down.

Dear Syntax,

I have been using my Adam for two years as a word processor for buisness. I am a freelance stage manager and lighting designer. Last summer, I stumbled upon a new use for the Adam and have used the low and high resolution graphics modes to create titles and graphics for three multi media shows. The graphics resolution of this machine is incredible when played back through a first rate 26" video monitor. I do not claim to be an expert (yet), but I would like to correspond with other Adam users with similar interests in theatre or multi media shows or graphics programs. My next step is to take a shot at sprites and I will keep you informed of my progress. The shows I work on seem to occur on short notice, but I will try to inform SYNTAX of any shows that Adam users may be interested in attending to see the Adam at work. The program on p. 4 is one that I use to tune and balance the color for a video monitor or TV. Using the square shape demonstrates any curvature of lines you may find on an older TV. Also because of the scan of the picture tube, color drawn vertically is not quite the same as the horizontal. HCOLOR 3-5 (white, black, medium red) are missing from this display, two are duplicates (3 = 7, 4 = 0) and 5 is close to 11.

Random notes of interest: 1. I have been using a modified version of "SHAPEMAKER" from Adam's Companion to define shapes for a shape table. The most useful modification has been a subroutine that allows you to change the scale, position, rotation or color of the shape as you build it. I also have this routine stored as a separate program for looking at shapes in a table. The routine also prints a hardcopy of SCALE, ROT, POS, COLOR and SHAPE # for future reference. 2. I have been working on a program that calculates throw distance and beam spread for theatrical lights and contains a database of over one hundred types of lights from North America and Europe. Currently, I am waiting to add one more manufacturer's data to my list. When this is complete, I can send a copy to SYNTAX if anyone is interested.

---

SYNTAX is published bimonthly by First Canadian Adam Users, Group (FCAUG), P.O. Box 547, Victoria Station, Westmount, Que., H3Z 2Y6  
Subscription Rates: 6 issues for \$22.00. Second Class Mail  
Registration No. 7152. POSTMASTER: Send address changes and notice of undelivered copies to above address. Return Postage Guaranteed. Copyright (c) 1986 by FCAUG. All rights reserved.

---

```

5 & Color Tune for Video Output
10 & Mr. Chris (Oct. 86)
20 & Square drawn from centre poker
30 Lomem :29000
40 POKE 16766, 96: POKE 16767, 109
50 size = 14
60 FOR address = 28000 TO 28000+size-1
70 READ byte%: POKE address, byte%
80 NEXT: REM shape data
90 DATA 01,00,04,00
100 DATA 18,63,36,36,45
110 DATA 45,54,54,63,00
120 REM draw color squares
130 HGR2: s = 1: ROT = 0
140 FOR c = 1 TO 15
150 FOR s = s TO s+2
160 IF c = 3 THEN c = 6
170 SCALE = s: HCOLOR = c
180 DRAW 1 AT 127, 95
190 NEXT s: s = s+1: NEXT c
200 GET key$: TEXT: END

```

Chris Walroth  
Toronto, Ont.

Adam's graphic capabilities are certainly one of its many great features. Besides being very entertaining, they can be very practical as you demonstrate with your work and enthusiasm. For more outstanding graphics, check out some of the programs in our BASIC LIBRARIES. Another small but useful graphics program which can be used for fine tuning the color on a TV or monitor appears in the programs section of this issue.

Dear FCAUG,

Another source for Adam ribbons is Data Ribbon Ltd., 130 Slater St., 2B, Ottawa, Ont., K1P 6E2 (613) 232-5728 . I live in the Ottawa area and have been going directly to them to get my ribbon cartridges refilled. The cost is \$4.00 per ribbon. Refills are usually ready for pick-up within 2 or 3 days. I have not had any problems with these refilled multistrike ribbons. The company will also accept for \$4.00 your used ribbon cartridge in exchange for a refilled one if they have available stock. The company advises they will cater to mail order requests for refills sent to the above address for the same \$4 fee per cartridge. Postage and handling charges are \$2.50 for orders up to and including 12 cartridges. Data Ribbon also sells new cartridges for about \$7.00 each. Ontario residents must add the applicable sales tax.

Maurice Forget  
Gatineau, Que.

Dear Editor,

I recently received my back issues of SYNTAX, and I must compliment you on filling a void. When I purchased my Adam, I also bought the SmartFILER software package. I had justified the

purchase (to my wife) on my need to catalog my record collection. I, then discovered (and you reported in your March/April 1985 issue) that the program had a limit of 256 entries. I managed to get a revised (revision 25, 7/23/84) tape by sending my old one to Coleco's Montreal office. I am still baffled after reading your report on SmartFILER. Surely, a filing system with only a capacity of 1000 will prove insufficient for many sophisticated collectors. My own record collection is quickly approaching 1000 and I don't want to waste my time inputting information to a system that will let me down at 1001! I wrote to Coleco with this question and have not received a reply. Perhaps someone in the group can recommend a solution.

Damian L. Bassett  
White Rock, B.C.

Don't hold your breath waiting for a reply from Coleco Canada. Writing to them is useless. All personnel that had any knowledge on the Adam has either left or was fired. Try calling them one day and asking about revision 27D. The stories and excuses you will get are worth the cost of the call. For more nonsense and total ignorance ask to speak to one of the managers. If we, as owners of one of their products living in the same city as them, cannot get proper advice and satisfaction from them, we shudder to think of the poor users living outside this area. Getting back to the question, depending on the length and size of your form, the maximum storing capacity on tape is 1000 records. With disks, it is even lower. As you can see this is related to the storage media (tape or single-sided disk). Since the program was designed to be used with either one of these, this seems to be the limit they can hold. It is important to understand that the limit is for a single tape or disk. You can work around this by dividing your database into subsections which is often safer and more practical. For example, in your case you can put all classical, jazz, rock, reggae, and country records on individual databases. If you prefer to file them in alphabetical order, you can break it down to A to L and M to Z databases. For most home management applications this is not a major inconvenience. Keep in mind that although SmartFILER is an excellent program, it is not dBase II or III, then again nor is its price. Back in March of 1985 when we printed our review on SmartFILER, we recommended that everyone who owned this software return it to Coleco Canada for revision 25 which was then the latest version. Afterwards, when SmartFILER was also released on disk, it was released as version 27D. The "D" apparently signifies Disk. We have not seen any noticeable differences between 25 and 27D, but it is always desirable to have the one with the higher number. At this point, we must caution you about sending in your rev 25 tape to Coleco asking for the updated 27D. Unlike many months ago when Coleco was still supporting the Adam, at this stage you not only risk losing your tape, but your temper as well. They refuse to handle anything dealing with the Adam. We would appreciate hearing from all those who do call Coleco to inquire about this and other Adam business. We will pass along your comments in upcoming issues. We will make 27D available to all those who send in their original SmartFILER

tape plus \$4 to cover return postage and labor.

Dear FCAUG,

I was very interested in your recent article on the DATA and REM bug. The three locations you suggest to poke (15824, 15830, 15831, 15832) with the new values are in block 17 at bytes 208, 214, 215, and 216 (D0, D6, D7, D8 in hex) respectively. Using a block editing program, I was able to modify a copy of SmartBASIC and eliminate the problem. This would appear to be an easier fix than modifying 25 bytes in block #2.

Bill Stewart  
Burlington, Ont.

Dear FCAUG,

SYNTAX 2.3 had a rather long fix for the DATA and REM bump problem. By modifying block #17, the new values for addresses 15824, 15830, 15831, 15832 can be made permanent on your BASIC tape or disk. While you are at it, you might want to change block #2, page 4, byte 57 to a NEW VALUE of 31. This will show Rev. 1.1 when you boot BASIC. Another optional change is to modify block #2, page 1, byte 04 to a NEW VALUE of 56. This will show revision year 86 when PRINT PEEK(260) is typed. All numbers are in hexadecimal.

```
BLOCK 17.....PAGE 1.....BYTE D0.....WAS D9.....CHANGE TO D8  
BLOCK 17.....PAGE 1.....BYTE D6.....WAS 13.....CHANGE TO 08  
BLOCK 17.....PAGE 1.....BYTE D7.....WAS 08.....CHANGE TO 37  
BLOCK 17.....PAGE 1.....BYTE D8.....WAS 37.....CHANGE TO 13
```

N.L. Durrant  
St. Catharines, Ont.

We are glad to see that many users picked up where we left off. As we mentioned, Bob's fix besides being lengthy, doesn't support "&" when used in place of REM. The alternative method is short, sweet and complete. We left it up to you to figure out which block and bytes to modify in order to write the necessary changes permanently to the tape or disk. Well done guys!

Dear Sirs,

The program "Gregorian Calendar" as it is on p. 20 of SYNTAX 2.3, will not print the 29th day for the month of February during leap years. To correct this, I have made the following changes. I also prefer to add line 67. It simply clears the screen when you ask for another month and year.

```
67 HOME: VTAB 7  
110 IF w = 0 AND z <> 0 THEN 112  
112 k = 29
```

Jean-Marc Roy  
Touraine, Que.

## Adam News and Developments

As promised in the last issue, FCAUG is now selling 64K memory expanders (\$75.00) and Centronics printer interfaces (\$85.00). Postage and handling charge is \$3.00. We hope to have 80-column cards available for sale early in the new year. We are planning to become a full service user group making it possible for you to obtain more Adam products in Canada. Write to us with your ideas for new programs or products that you would like to see developed for the Adam.

R.W. Greene of Kelowna, B.C., informs us that he has received correspondence from an American source marketing an Adam-IBM combination package. "All the drives are immediately converted to respond to either Adam commands or IBM commands, and respond to MS-DOS commands as well." It is apparently text compatible with IBM and will run many IBM programs except those that are hardware dependent. Price tag - \$599.95 + \$15.00 U.S. With exchange and federal sales tax that makes the Canadian price about \$893.00.

David Allen of Beamsville, Ont., tells us that Loranger (the manufacturer of the original Coleco data packs) will continue producing data packs under the LORAN label. You can special order them through most Canadian Tire outlets. The product number is 69-3024-4 and the price is \$7.99 each.

FCAUG member Gerry Totten and several others in Regina, Sask., have started a local users' group. He mentions that after their initial meeting, they already had more than 25 members. Address: Regina Adam Users' Group, c/o Gerry Totten, 59 Woodsworth Cres., Regina, Sask., S4T 7A9 Tel. (306) 545-9309

Another FCAUG member Jason Rheindel has launched his own users' group. Write to him and receive a free collection of educational and entertaining programs. Send \$1 to cover postage. The address: ADAMania, Box 269, Lampman, Sask., S0C 1N0

The Adam Evaluation Club has moved to the following new address: RR #2, Bolton Pass Rd., Knowlton, Que., JOE 1V0 Tel(514) 243-6461

Due to poor local support in the Kansas area, K.A.U.G. will cease to publish its fine newsletter. We wish to congratulate and thank David Carmichael, Lewis Clancy and others for the hard work and dedication they demonstrate in supporting the Adam computer. Your efforts have not gone unnoticed by all Adam users.

Anything For A Buck Dept.

An American supplier, called Marathon Computer Press, is offering to correct for a \$5 fee, the DATA and REM bug on your SmartBASIC. This is the most ridiculous thing we have seen. As pointed out in SYNTAX 2.3, users unfamiliar with block editing programs to make the necessary changes themselves can simply put the four required pokes in an HELLO file on their SmartBASIC. On boot-up, this will have the same effect as modifying the bytes permanently.

The Forty Column Text Mode

The video display processor (VDP) in the Adam has four kinds of display modes. They are the Graphics, Bit-mapped, Multicolor, and Text modes. Surprisingly, SmartBASIC only makes use of the first two modes. In this article, I will introduce you to all four VDP display modes and include a short BASIC program which gives you access to the VDP's 40 column Text mode.

SmartBASIC uses the VDP's Graphics mode as its TEXT mode. This mode displays 32 columns by 24 rows of graphic tiles. A graphic tile is an 8 by 8 dot matrix which takes 8 bytes to define its pattern. Since these graphic tiles are usually defined as text characters, we often refer to them just as characters. We have 256 characters to play with. Groups of 8 characters will have a common color byte. A color byte specifies the group foreground ("on" dots) color in the most significant 4 bits and the background ("off" dots) color in the least significant 4 bits. SmartBASIC makes the first and last 128 character patterns the same except reverses the foreground and background colors. This allows for FLASHing, INVERSE, and NORMAL characters to be displayed.

The VDP's Bit-mapped mode is used for SmartBASIC's Lo-Res and Hi-Res graphic modes. In this mode, there is a separate graphic tile for all of the 32 by 24 screen locations. This allows for all of the dots (pixels) on the screen to be set independently. Rather than having a color byte for each pixel, there is only one for a group of four. This means that only two colors can exist in a 4-pixel color group and this is the reason why colors tend to spill over or "bleed" in the Hi-Res modes.

SmartBASIC doesn't make use of the VDP's Multicolor mode. In this mode you have 64 columns by 48 rows of 4 by 4 dot blocks. Each block can be any color just as in SmartBASIC's Lo-Res modes. Playing with colored blocks has very limited use consequently SmartBASIC's exclusion of this mode is no great loss.

On the other hand, the exclusion of the VDP's Text mode by SmartBASIC is a great loss. This mode allows 40 columns by 24 rows of text. It can use the graphic patterns as defined in the VDP's Graphic mode but ignores the last 2 columns of dots. This means a text character can only be 6 dots wide and 8 dots high. Since the text characters in the Adam's ROM is defined in a 5 by 7 matrix, we can use them in the VDP's Text mode without any problems. The only problems we have is with the Adam's graphic characters. They are wider than 6 dots and will have their right-most dots chopped off when displayed in the VDP's Text mode. As an example, try typing in the command: PRINT CHR\$(5), you should see a white diamond. After running the BASIC program to change to the 40 column text mode you will find a chopped diamond printed. A disadvantage of the 40 column text mode is that all of the characters must have the same foreground and background colors.



As a result, you can not get inverse or flashing characters in this mode. Another disadvantage is that there are no sprites available in this mode, while in all the other modes 32 sprites are available.

The VDP has 8 write-only registers which control the way it works. By setting certain bits in these registers, we can change display modes as well as arrange the location of tables in the VDP's RAM. For the VDP's Text mode we will only concern ourselves with registers 1 and 7. To select the VDP text mode, we send a value of F0 hex (240 decimal) to register 1. Register 7 is used to set the foreground and background colors for all of the text characters.

The BASIC program included on p. 19 will change BASIC's TEXT mode to make it display the VDP's 40 column Text mode. This requires changing some of SmartBASIC's code. You will have all the normal TEXT commands at your disposal including the POKE at location 17115 which changes the text colors. I will only give a cursory description of the various POKES the program is making. For a fuller understanding, I suggest you buy "The Hacker's Guide to Adam - volume two" by Ben Hinkle (117 Northview Rd. Ithaca, NY 14850). This guide gives a very detailed description of the inner workings of SmartBASIC and contains some interesting material that I hope to make use of in future articles.

Most of the BASIC program is concerned with modifying various addresses of the BASIC interpreter to accomodate a 40 column screen. To change to the VDP Text mode you only need to change VDP registers 1 and 7. You can do this with the following assembly code.

```
LD BC,$07F0      register B = VDP register 7
                  register C = data for VDP register (F0)
CALL $FD20       system routine to change VDP registers
LD BC $01F0      register B = VDP register 1
                  register C = data for VDP register
CALL $FD20
```

The first CALL changes the text character colors to white on black, the second call puts the VDP into Text mode. Lines 40 to 70 of the program are concerned with patching the TEXT initialization routine. Lines 90 and 100 POKE a routine to help calculate the new screen offset. Lines 120 to 140 let the BASIC routines know of the new location of the screen line buffer. Normally, SmartBASIC uses a 32 byte long buffer located at location 16960 to read from and write to the screen. But since we are using a 40 character line we need a 40 byte long buffer. This new buffer is placed at location 28094.

Incorporating this BASIC program in one of your own programs will give you the ability to use a larger text display. For programs that are very text oriented, this will add a very nice touch. I hope you will try using it in one of your future programs and find it useful.

## Machine Language Primer

In the following article, I will show you how to use the machine code - mnemonic table presented in the last issue of SYNTAX by going through the steps involved in writing a machine code routine.

We will use SmartBASIC to put our machine code routine into memory and to call it. To do this, we must first set aside some reserve memory. We do this by using the LOMEM or HIMEM statement. The LOMEM:NN statement will reserve memory from location 27408 to NN. The HIMEM:NN statement will reserve from NN to 53631. In most cases, we reserve a chunk of memory that is much larger than the routines we wish to place in it. This allows for some breathing space in case we expand our routines.

We use the POKE statement to place our machine code routine into the reserved memory. Most of the time we will use DATA statements to hold the machine code bytes. We will then read and poke each byte into memory. After the routine is in memory, we will CALL it. The CALL statement works just like the GOSUB statement except it jumps to a memory location and executes machine code. When the machine code ends it will return to the BASIC program. Let's go through an example to illustrate this.

There are many routines in the SmartBASIC environment that we can make good use of. This includes routines inside the SmartBASIC interpreter as well as the computer operating system (EOS). We will make use of a routine inside the SmartBASIC interpreter which displays a character on the screen. This routine expects the A register to contain the ASCII value of the character to be displayed. Here is the assembly routine to print the character "A" on the screen.

```
LD A,41h      register A = 41 hex (ASCII for "A")
CALL 2F0Bh    call the routine to print a character
RET          return to BASIC
```

As you can see, this requires very few instructions. It simply requires you to load register A with an ASCII value, call the appropriate system routine, and return to BASIC. The CALL opcode in assembly language works just like the one in SmartBASIC. All the messy stuff has been done in the mysterious system routine. As a machine code programmer this is a blessing since you would not want to have to code all those extra bytes. The best way to approach system routines is to view them as black boxes. You need to know what the routine does, what registers it requires before calling it, and what registers it might return information in. You do not need to know its inner workings.

Now that we have the assembly language routine, we need to convert it to machine code bytes. To do this, we use the machine code - mnemonics table on pages 18-19 of SYNTAX 2.3. We start by looking for the LD A,N instruction. When we find it, we see the

hex numbers 3E20 to its left. The 20 hex (second hex byte) represents a dummy data value, N. The designers of this table chose N to be 20 hex. In our case, we have N equal to 41 hex, therefore the machine codes we need for LD A,41 are 3E 41. Next, we search the table for the CALL NN opcode. We find it has the machine codes CD8405. The underlined numbers is where we place our address that we are going to jump to. The designers of the table again chose some dummy value to represent the two byte constant NN. As usual with Z80 word representation (a word is 2 bytes), we must store our address in reverse byte order. Therefore, the machine codes for CALL 2F0Bh are CD 0B 2F. Our last opcode is the RET instruction. This instruction is just like the RETURN instruction in BASIC. We search the table and find the machine code C9 to its left.

As you can see from this example, an opcode can have a variable number of bytes to represent it in machine language. If we wish to use these machine codes in our BASIC program, we need to convert them to decimal. We can use a hex-to-decimal calculator or a short BASIC program. Here is a simple, "one-liner" hex to decimal conversion program that you can type in. Make sure you type it exactly as printed here since it approaches the maximum input length in SmartBASIC. The program expects a 2-digit hex number and will display its decimal count- erpart. Make sure you type a true hex number (the hex digits are 0,1,2, ... 9,A,B,C, ... F) since the program is too short to check for this itself. To end the program hit Control-C followed by the return key.

```
1 INPUT "Hex # ";h$: FOR i=1 TO 2: a=ASC(MID$(h$,i,1)):
n=n+16^(2-i)*(a-(a<58)*48-(a>96)*87): NEXT: ?"=> ";n: RUN
```

Converting the hex machine codes to decimal gives us the DATA statement shown in line 120 of the program. In this program (see p. 19), we reserve memory with the LOMEM:28000 statement. Line 110 of the program loads the print character routine starting at 27500. It goes and pokes all 6 bytes into memory. To see an "A" on the screen, we CALL 27500 as in line 130.

If we want to display another character, we only need to change one byte of the machine code routine. This is the second byte which is the data value to load into register A. When the routine is loaded into memory, we can POKE in a new ASCII value by simply POKEing to location 27501. Hence, it is as if we are loading register A from BASIC. Line 150 pokes the ASCII value 12 (control-L) into the second byte of the machine code routine. This will clear the screen when the routine is called. Line 170 pokes a 2 (control-B) into the routine so that it can print a bunch of happy faces. See the ASCII table in appendix C of your SmartBASIC manual for more ASCII codes.

As you can see this routine is not all that useful in SmartBASIC since a PRINT statement does a better job. But it does instruct you on how to code a machine language routine and how to use it in BASIC. Next time, we'll explore many more interesting system routines.

## E-Z EDIT

E-Z EDIT is FCAUG's companion program to E-Z COPY. If you have used and are familiar with E-Z COPY, you should have no trouble using E-Z EDIT. This is (of course) intentional. For ease-of-use, the look of the two programs have been kept the same from the welcome screen, to the select drive screen, to the select block screen. But instead of copying many blocks, E-Z EDIT copies one block allowing you to make changes to it before you write it back again.

This is how it works. You load a block into memory. Next, you are prompted to decide how you wish to see the block to be edited. You have two choices. You can either see the bytes expressed as decimal values or hexadecimal and ASCII values. If you don't know what this is by now, I will briefly explain it. Computers store information in ways that can be easily represented by 0's and 1's (binary number system). This same information can be expressed under the decimal (0-9) system or the hexadecimal (0-15) number system. Sort of like the way things that most of us wear on our feet are called shoes in English and "souliers" in French - meaning that a shoe is a shoe, no matter what we call it. Understanding this is important because it allows us several very convenient ways to work with computer systems. "ASCII" is another useful standard. It is a pre-arranged code that says a certain permutation of these 0's and 1's represent a certain character. Basically, the display that you choose is just a different way of representing the same thing (value of a memory address). We will get to how these numbers are used later.

Another concept to grasp is that of the "buffer". This is simply a reserved area in memory used to put specific information. In our case, it's 1K worth of information from a tape or disk. Once you load a block in, you can then make as many changes as you like to to the bytes stored in your buffer. If you want to edit another block, it too is written to the buffer, but any previous information will be overwritten and lost if not saved to your media.

Now that you know what a buffer is and how the information in that buffer may be displayed, let's see what E-Z EDIT can do. If you choose the decimal display, the addresses and the contents of your buffer are represented as decimal numbers. A menu screen offers you the choice of changing the value of the address of where the cursor lies, moving to another address or exiting the edit buffer mode. There are 2 numbers to be aware of on the screen. One, the present "offset address" will always be 0 when you first enter the edit mode. The other is the value of the present offset address. The highest "offset" address will be 1023. Offset means how many places you are away from the start of the block (so the first byte at offset 0 is 0 places away from the start or in other words not away from the start at all). This means that each byte in a 1K block can be represented on the screen. The values will be anywhere from 0 to 255.

Once you get to the byte you want to change, you simply type your new byte value in and continue with all the changes that you wish to make in that block. The arrow keys are used to move to successive addresses in the buffer. The right arrow moves one byte forward, the left arrow one byte back. The up arrow, 10 bytes forward and the down arrow, 10 bytes back. The <CONTROL> key and the right arrow key moves 100 bytes forward, <CONTROL> left moves you 100 bytes back. <CONTROL> up moves you to the beginning of the buffer and <CONTROL> down moves you to the last address, offset 1023 in the buffer.

If you use the other display format, hex/ASCII, you will have a more sophisticated way to look at your buffer. Instead of 1 byte being displayed on the screen at a time, you get a whole screen full or 128 bytes. The present address is displayed at the top of the screen. The values are displayed below in hex on the left part of the screen and ASCII on the right. Arrow keys are again used to control which address you wish to get to. The <CONTROL> down key puts you into the next screen and the <CONTROL> up key puts you one screen in reverse. <CONTROL> left will return you to the first screen and <CONTROL> right will put you at the last screen. The <TAB> is a toggle that puts the cursor from the hex display part of the screen to the ASCII part and back again. Writing over a value replaces the old value with the newly entered one. Each entry automatically updates the screen.

You can exit both edit modes at any time. If you decide to do this, you can then exit the program or choose to edit a new block. Once you have edited a block, you are given a second chance to choose a different drive and/or block to write to than the one that you chose at the beginning. This gives the program added flexibility.

By knowing the block number and the offset of the byte that you want to change you can make any change to a disk or tape that you want. This gives you a very powerful tool to customize your own software in any way that you wish. E-Z EDIT also allows you to look through blocks and see without changing them what is really stored on your media. SYNTAX has already mentioned some useful CP/M changes as well as several SmartBASIC changes that you can make with a disk or tape editing program such as E-Z EDIT. Look for other interesting applications for this program from us in the near future.

---

#### USERS' CORNER

Daryl Heinsohn  
72 Houle, Box 211  
Dowling, Ont.  
POM 1R0

Guy-R. La Forest  
P.O. Box 548  
St. Jacques, N.B.  
EOL 1K0

T.M. Macsymic  
293 Riverside Dr.  
Prince Albert, Sask.  
S6V 2W9

Review of the Adam Disk Drive  
by Greg Slade

If you have been considering digging deep into your pockets to invest in a disk drive, but haven't been sure if it was worth the extra money, let me give you a little encouragement. When I first unpacked my shiny new disk drive, I was amazed at how much more versatile and powerful my system had become. My trusty little V.W. had become a Porsche. (Well, okay, Fiero then.) I timed both the digital data drive that came with the system and my new disk drive in storing and verifying the same SmartWRITER document. The data drive took 59 seconds, not bad, especially when compared to a "normal" tape drive like Commodore's. The disk drive managed to slice that time into 18 seconds! (I'm thinking of putting racing stripes on it.)

You needn't worry about learning some complicated procedure for using disks. Once you've set your baby up and turned it on, you follow simple directions to load the Adam Disk Manager (the Disk Operating System or DOS) into Adam. Then, after "backing it up" onto the free disk which comes with the drive, you can format as many disks as you're going to use. (See SYNTAX 2.2, pp. 8-9 for more details on the Disk Manager and using a disk drive.) I find it easiest to backup the DOS onto my SmarWRITER file disks. It uses up 25K, but I have instant access to my files that way.

If you find yourself twiddling your thumbs while Adam STORE/GETs a file, consider a disk drive. Although a disk drive for the Adam is hard to locate these days, and most often costs the price of the computer alone or more, it is worth every penny. Once you have acquired one, you will have made the most important system upgrade and you will soon be wondering how you ever worked with those slow tapes before. One more thing, even though disks only hold 160K as opposed to 256K on a Digital Data Pack, you can get ten disks for about the price of two data packs.

-----  
TECHNICAL SPECIFICATIONS

Format: 5-1/4" Single Side, Double Density  
Transfer Rate: 24,000 bits per second  
Total Storage: 160K bytes  
Track Density: 48 TPI  
Tracks: 40  
Recording Density: 5536 (BPI)  
Sectors: 8 per track, 512 bytes per sector, 320 sectors per disk  
Access Time: Track to Track 20 milliseconds  
Rotational Speed: 300 + 1-1/2% (RPM)  
Heads: tunnel-erase manganese zinc heads  
L.E.D. Indicators: Two, Drive In Use, Power On  
Drive Select Switch: Two position  
Power Requirements: 9 volts a.c., 3.4 amps (transformer supplied)  
Multiple Drive Capabilities via connectors on back of drive  
Dimensions: 1.625"H x 5.75"W x 9"D

## Program Review: ExpertType - Coleco Industries Inc. (1984)

As computer users we all have to use the keyboard to enter our input be it text or programming instructions. This requires us to use our typing skills. If we can learn to do this quickly and accurately, we will simplify our task and become more efficient users. ExpertType is a program designed to help us (no matter what age or ability) reach this goal by teaching us the touch-typing method. Touch typing teaches us the correct fingers to use for each key so we can type without looking at the keyboard.

ExpertType is a custom-designed typing instruction program. It creates individualized typing lessons based on a profile it keeps of every user. Each user profile contains name, age, typing speed and accuracy, a list of all exercises completed and a list of problem keys. Each time you end your typing session, your profile is updated and stored on the data pack or disk when you press the SIGN OFF key.

Each of the 12 touch-typing lessons teaches correct finger placement on the keys, and provides many exercises that introduce new characters for you to practise. Your instructions regarding what finger to use for each key on the keyboard and what to type are given throughout the lessons. You must type exactly what appears on the screen - characters and spacing. As you complete one line, that line moves above the roller area and Adam underlines your mistakes in red. You cannot go back and correct them. It is important to continue typing as your speed is being calculated. The program will show you your problem keys later on. After you have completed five touch-typing lessons, Adam will automatically give you a report card. Your typing speed and the keys you need to practise will be displayed. You can select another kind of typing lesson by going to the Select Lesson screen anytime.

There are other lessons built around characters and words as well as free-style and dictation lessons. Character lessons help you practise your problem keys Adam has determined by calculating your speed and accuracy with each key. Word lessons feature those characters that you need to practise. In free-style, you can type randomly from the top of your head or let Adam provide you with a thought which you are asked to complete. With dictation, you can practise transcribing text which appears at the top of the screen or from printed output which you can have the program print out. After every free-style or dictation lesson, Adam will display a keyboard graphic on-screen. Your problem keys will be highlighted in red. The next new lessons you select will be especially tailored to emphasize your problem keys.

Every thing in this program is well done from the variety and flexibility of the exercises to the presentation and ease of use. This is a very complete and useful program to have if you are serious about learning to type properly with reasonable speed and accuracy. If you are content to continue hacking away using the "hunt and peck" method then you will find little value from a program of this type.

## Program Review: Smart LETTERS & FORMS - Coleco Industries Inc.

Here is an interesting software package which is extremely easy to use. Although the collection of letters is quite extensive, the question of just how practical are these examples remains an individual decision. If you have difficulty composing letters, you might find some use for this program. On the other hand, if you are the creative type this is definitely not for you.

This program operates strictly in the moving window format. For those of you who are accustomed to working in the standard format, a little re-orientation might be in order. Remember that your screen is now a window over the text. You can scroll to text above, below and to either left or right of the screen. This is preferable so that when working with a selected form you can quickly scan across it to fill in the contents of the letter. It also avoids working with a cluttered screen.

The program's built-in word processing features make text editing easy. You have at your disposal all of the SmartWRITER functions except MOVE/COPY and CLEAR. Margin settings and line spacing cannot be altered. If this poses a problem you can store the form and edit it later using the word processor. A new function, CENTER, is introduced. Any line can be easily centered both on the screen and on paper by simply placing the cursor anywhere on that line and hitting the CENTER smart key.

At the entry level screen the smart keys display four options: Social, Personal Business, Check Lists, Misc, and Story. This last option constructs a story out of the various sample letters. To see the directory of one of the four categories you press its smart key. Under Personal Business for example, are included letters of employment, complaints, information request, appointments, accommodations and general. The form of the letter you select will appear consisting of text and highlighted labels. A sample of the selected letter is also included. To compose your letter you insert information in the labels using the TAB key. This brings you to the next label on the form and automatically puts you in the INSERT mode. You can also use the arrow keys to move your cursor to any part of the label and use the INSERT key. Using the TAB key is preferable as it saves you scrolling time in locating the next label to be filled in. In the INSERT mode, the label is shifted above your workspace to remind you of the type of information you should enter. Once your information is typed in, you hit the DONE smart key and TAB over to the next label. If you store the form you can also merge it with your SmartFILER or ADDRESS BOOK databases. The disk version of this program is divided into two disks to compensate for their reduced storing capacity. What this means is that it is quite difficult to transfer this program to disk if you have the tape version.

If you take this program and look at its directory in the W/P mode, you will discover an interesting aspect. What you will notice is that its directory holds more than the limit of 35 files. It actually holds a total of 101 files (on data pack) indicating the use of a multi block directory.



Program Review: The Best Of Electronic Arts - 1984  
Hard Hat Mack and Pinball Construction

In real life, construction work can be very laborious. Hard Hat Mack captures this aspect in a challenging and creative game based around a construction site quest. In each of the three screens, there is a job to be done and it must be done fast and right in order to remain on the job.

In the first screen there is a series of horizontal steel beams that need repair. The only means of moving from one level to the next is through the freight elevator on the left, the springboard on the right or the ladders. You must direct your Hard Hat Mack to repair the gaps in the beams. To do this, Mack must transport the missing section and then bolt it in place with a jackhammer. Mack can grab the jackhammer at random as it moves about the beams. To complicate life for Mack, there are obstacles and wild lunatic characters which pose a constant menace. Mack must labor twice as hard to stay out of their path. Then there is the bouncing spike which is released from the top right hand side diagonally across the screen. Not even a hard hat offers protection against this type of hazard. Speed and agility are Mack's only defense against these deterrents.

The second screen tests Mack's nerves of steel. He must be moved patiently onto a conveyor belt while dodging huge cement-like drops to ride a beam being lifted by a crane. By jumping back and forth between the stationary beams and the moving one, he must retrieve six tool bags. The major obstacles here are; a pounding hammer, dynamite, a moving vise, and the ever-present lunatics. The objective in this screen is to get to the crane. But to reach it, Mack must overcome yet another conveyor on the upper tier - a very difficult feat to accomplish.

In addition to excellent coordination and quick reflexes, screen three calls for some good imagination. There are two springboards plus a large vertical conveyor and a smaller horizontal one which Mack can use to move around. Imagination comes into play not only to gather the boxes, but also in avoiding the lunatics in order to bring them to their destination. A pounding hammer and flames on the horizontal conveyor (where one of the boxes is located) spell danger. Mack's previously acquired skills are put to the ultimate test here to complete this level.

Hard Hat Mack is a very demanding game. It is similar to Miner 49er and just as tough. Most of the tasks to be accomplished are difficult, but the attempt can be frustrating and fun at the same time. It will take considerable practice to master this trade!

Also included in this two-game package is Pinball Construction. This program consists of tools and components (e.g. flippers, bumpers, etc.) for you to use to create your own pinball games. Two demo games showing the many possibilities are provided. With the many gadgets to choose from, each person can create and save personalized pinball games independent of age or ability.

```

10 & * HGR2 BOUNCE *
12 DIM g(2), inc(2), x(2), xt(2), y(2), yt(2)
14 HGR2: HCOLOR = 3
16 HPLOT 0, 0 TO 255, 0 TO 255, 191
18 HPLOT TO 0, 191 TO 0, 191 TO 0, 0
20 FOR i = 1 TO 2: inc(i) = RND(1)*4+2
22 g(i) = 0: x(i) = 0: NEXT: y(1) = RND(1)*40
24 y(2) = RND(1)*70+10: c = 0
26 FOR i = 1 TO 2
28 g(i) = g(i)+.9: x(i) = x(i)+inc(i)
30 y(i) = y(i)+g(i): IF y(i) <= 190 THEN 34
32 y(i) = 190: g(i) = -g(i)/1.05
34 xt(i) = x(i)
36 IF x(i) > 254 THEN xt(i) = 254: GOSUB 50
38 yt(i) = y(i): NEXT i
40 IF x(1) > 254 AND x(2) > 254 THEN 48
42 IF c = 0 OR y(1) = y(2) THEN 46
44 yt(c) = y(c)+(y(3-c)-y(c))*h2/ABS(x(1)-x(2))
46 HPLOT xt(1), yt(1) TO xt(2), yt(2): GOTO 26
48 FOR d = 1 TO 900: NEXT: GOTO 14
50 h2 = x(i)-254: c = i: RETURN

```

```

10 & * COLORTEST *
15 a = 0
20 b = 16
25 c = 0
30 HGR
35 FOR x = a TO b
40 IF x >= 255
    THEN 65
45 HCOLOR = c
50 HPLOT x, 0
55 HPLOT TO x, 160
60 NEXT
65 c = c+1
70 a = (c-1)*16+16
75 b = c*16+16
80 ? TAB ((c-1)*2);
    c-1;
85 IF x >= 255
    THEN END
90 GOTO 35

```

```

-----
10 & * STARCITY *
12 HGR: HCOLOR = 3
14 FOR k = 1 TO 35: n = 36
16 a = INT(26*RND(n))
18 HPLOT a, a+k
20 HPLOT 10+a, 20+a+k
22 HPLOT 200-a, 150-k
24 HPLOT 3*a, 3*k
26 HPLOT 4*a, 4*k
28 HPLOT 240-4*a, 150-3*k: NEXT
30 FOR c = 0 TO 250
32 a = INT(36*RND(n))
34 HPLOT 2+c, 150 TO 2+c, 145-a
36 NEXT c
38 FOR c = 1 TO 100
40 SCALE = 1
42 DRAW 1 AT 15+c, 15+c
44 ROT = 2: ROT = 3: ROT = 3
46 XDRAW 1 AT 15+c, 15+c
48 ROT = 2: ROT = 3: ROT = 3
50 NEXT c
52 FOR s = 1 TO 35
54 LET a = INT(26*RND(n))
56 HPLOT 150, 150 TO 70-a, 135-a
58 HPLOT 150, 150 TO 70+a, 127-a
60 HPLOT 150, 150 TO 150+a, 150-a
62 HPLOT 150, 150 TO 200-a, 150-a
65 NEXT s
70 DRAW 1 AT 115, 115
75 FOR a = 1 TO 100
80 DRAW 1 AT 115-a, 115-a
85 ROT = 6
90 XDRAW 1 AT 115-a, 115-a
95 ROT = 6
99 NEXT a

```

```

-----
10 & BAR CHART for printer
12 & Eric Lauzon (1986)
14 ? : ? TAB(5);
16 ? "% d'accomplissement en 1986"
18 ? : ? TAB(5);
20 ? "000.....050.....100"
22 FOR nxc = 1 TO 12 : READ m$, p
24 ? TAB(5); "|"
26 FOR b = 1 TO 3
28 IF b=2 THEN ? m$; "+";:GOTO 32
30 ? SPC(5);
32 FOR a = 1 TO p*.25
34 ? "["; : NEXT: ? : NEXT: NEXT
36 ? TAB(5); "|": ?
38 ? : ? TAB(6);
40 ? "MOIS"; TAB(18); "???: ?
42 RESTORE
44 FOR nxc = 1 TO 12: READ m$, p
46 ? TAB(6); m$; TAB(18); p
48 NEXT: END
50 DATA JAN., 5,FEV.,10,MAR.,15
52 DATA AVR.,20,MAI.,25,JUN.,30
54 DATA JUL.,50,AOUT,65,SEP.,75
56 DATA OCT.,80,NOV.,90,DEC.,100

```

```

-----
10 GR: & * LOW-RES DEMO *
15 COLOR = 14: GOSUB 45
20 COLOR = RND (1)*15+1
25 HLINE 0, 39 AT RND(1)*40
30 COLOR = RND(16)
35 VLINE 0, 39 AT RND(1)*40
40 GOTO 20
45 FOR j=0 TO 39:VLINE 0, 39 AT j
50 NEXT: RETURN

```

```

10 & **40 column text mode**
20 LOMEM :28400
30 & machine code to switch to the VDP TEXT mode
40 DATA 1,240,7,205,32,253,24,14
50 FOR x = 0 TO 7: READ d: POKE 17114+x, d: NEXT
60 POKE 17166, 192: POKE 17177, 192: POKE 17199, 39
70 POKE 17215, 240: POKE 17988, 40
80 & change screen offset routine
90 DATA 41,197,229,41,41,193,58,112,66,183,32,1,9,193,201
100 FOR x = 0 TO 14: READ d: POKE 16976+x, d: NEXT
110 & change these addresses to reflect the new line buffer address
120 DATA 17985,18036,18098,18162,18174,18188,18401,18410,18430
130 FOR x = 1 TO 9: READ y: POKE y, 190: POKE y+1, 109: NEXT
140 POKE 18272, 205: POKE 18273, 80: POKE 18274, 66
150 TEXT

```

-----

```

90 & **Machine Language Primer Program**
100 LOMEM :28000: TEXT
110 FOR i = 0 TO 5: READ d: POKE 27500+i, d: NEXT
120 DATA 62,65,205,11,47,201: & print character routine
130 CALL 27500
140 FOR delay = 1 TO 1000: NEXT: & delay to show "A" ON SCREEN
150 POKE 27501, 12: & load register A with 12
160 CALL 27500: & clear screen
170 POKE 27501, 2
180 FOR i = 1 TO 100: CALL 27500: NEXT: & print 100 happy faces

```

-----

```

10 & **SNOWFLAKES**
12 &
14 & J.M. Roy (1985)
16 ONERR GOTO 66
18 HOME
20 VTAB 5: HTAB 11: ? "SNOWFLAKES"
22 VTAB 7: HTAB 15: ? "by"
24 VTAB 9: HTAB 12: ? "J.M. Roy"
26 VTAB 20
28 ? "Enter # (1 to 9)": GET a
30 a = RND(-a)
32 HGR2
34 FOR n = 1 TO 15
36 IF n = 4 THEN NEXT n
38 HCOLOR = n
40 x = INT(RND(1)*205+25)
42 y = INT(RND(1)*130+25)
44 z = INT(RND(1)*20+5)
46 HPLOT x-z, y TO x+z, y
48 HPLOT x, y-z TO x, y+z
50 z = z-3
52 sx = x-z: sy = y-z
54 HPLOT sx, sy TO sx+2*z, sy+2*z
56 sy = y+z
58 HPLOT sx, sy TO sx+2*z, sy-2*z
60 NEXT n
62 FOR t = 1 TO 1000: NEXT t
64 GOTO 32
66 TEXT: END

```

```

10 & **ASCII slant**
12 &
14 & J.M. Roy (1986)
16 ONERR GOTO 56
18 POKE 16953, 32
20 t = 0
22 t = t+1
24 IF t=7 OR t=8 OR t=9
   OR t=10 OR t=12 THEN 22
26 IF t=13 OR t=16 OR t=19
   OR t=22 OR t=24 THEN 22
28 IF t=28 OR t=32 OR t=127
   OR t=128 OR t=148 THEN 22
30 IF t=151 OR t=160 OR t=161
   OR t=162 OR t=163 THEN 22
32 IF t > 255 THEN 56
34 HOME
36 ? TAB(10); "WATCH THIS!"
38 VTAB 8
40 ? "CHR$( "; t; " )="; CHR$(t)
42 FOR x = 1 TO 2000
44 NEXT x
46 HOME
48 FOR c = 1 TO 23
50 ? TAB(c); CHR$(t)
52 NEXT c
54 GOTO 22
56 POKE 16953, 95: HOME

```

----- F C A U G    P R O D U C T   L I S T   -----

E-Z EDIT - \*\* NEW \*\* An alternative tape/disk editing program that is easy to understand and use. Price \$15. [ ]

E-Z COPY - The utility program that lets you copy only the blocks you need. Save time by not copying empty blocks. Make back up copies of all your valuable software. Price \$15. [ ]

FCAUG BASIC PROGRAM LIBRARY #1 - A collection of 30 programs with games, graphics, sound, home management applications. \$10. [ ]

FCAUG BASIC PROGRAM LIBRARIES #2 & #3 - Even more programs demonstrating the infinite capabilities of the ADAM. Some really superb graphics and games programs! Price \$10 each. [ ]

CP/M 2.2 COMMUNICATIONS PACKAGE - Includes MEX, ADAMBOOT and two versions of MODEM 7. These are powerful and versatile programs useful for transmitting binary files. Price \$10. [ ]

PUBLIC DOMAIN PROGRAMS - \*\* NEW \*\* Quality programs that were never released commercially by Coleco and have been placed in the public domain. Included are Troll's Tale, Super Subroc, Jeopardy, Hard Hat Mack/Pinball Construction, SmartBASIC 2.0. \$10 each [ ]

DATA DRIVE SPEED TESTER - Verify and adjust the correct speed of your data drive(s). Instructions included. Price \$10. [ ]

Now available through FCAUG at a new low price - APE Software's ELECTRONIC GAME PACK - Classic strategy games like Mastermind, Battleship, Backgammon, 3-D Tic-Tac-Toe or Robot Miner. \$20. [ ]

ARCADE PACK - \*\* NEW \*\* Just released by APE, 4 machine language games. Sky King - defend against enemy forces with your helipod. Car Wars - outrace and outwit your competition. Crater Runner - mine astro craters for precious gems. Snake Weeds - destroy these advancing and spreading pests with your laser cannon. Order from APE or through FCAUG. Add \$2 for postage. Price \$20. [ ]

Specialized new programs written by Chris Hills. BSR - a "block transmitting" modem program capable of sending the complete contents of any disk. TRS-2-ADAM & TRS-DIR - Read TRS directories to convert TRS CP/M files for use on Adam. TIMEX - time display with "multi-tasking" feature. Price \$20 each TIMEX \$10. [ ]

Centronics (Parallel) Printer Interface card    Price \$85.    [ ]  
64K memory expansion card                            Price \$75.    [ ]  
SmartBASIC 1.0 on ROM cartridge.                    Price \$70.    [ ]

Now load SmartBASIC super fast from game slot. Numerous bugs have been fixed. Extended features to be added if demand is strong. Future upgrades provided to original buyers at minimal cost. For hardware items, please add \$3 for postage and handling.

FCAUG    Box 547, Victoria Station, Westmount, Que. H3Z 2Y6