



THE EASY WAY

By Dennis Lewandowski

Radio Shack released a Color Computer with 4K, expandable to 16K. That was it; 16K was to be the maximum amount of memory that could be obtained from it. However a rather intelligent gentleman wrote an article in the March 1981 *BYTE* magazine, telling how to piggyback 16K chips and fool the SAM and CPU into seeing 32K.

Rumors flew fast and furious about a 32K memory kit from Radio Shack. It involved the use of partially bad 64K chips. Now the minds at Frank Hogg Laboratories went to work. Using an extra gate already available on the board, they could toggle the upper bank of the 64K chips in and out, thus, making the current 64K Color Computer we have today. Everyone went to work to develop a 128K modification, but the same stumbling block kept getting in the way; the ROM version of BASIC will only support 32K. With prices dropping on computers faster than pig bellies on Wall Street, most research went the way of the horse. Sure, there are a couple of 128K modifications currently available, however the hows and whys are being kept guarded secrets, making software support almost impossible.

(Dennis Lewandowski, one of the early authors active with the Color Computer, specializes in machine language programming. He and his wife, Rose, founded DSL Computer Products.)

Now a little background on this 128K modification. The main objectives are basically common sense. The modification must be usable by BASIC, and relatively inexpensive to upgrade the current CoCo. It also has to like FLEX, and OS-9. For these objectives to be reached we chose to use two sets of 64K chips. There are six other chips involved in the modification, which take care of selecting the banks. Actually five of the chips take care of the bank selection, but due to timing considerations, the sixth chip makes certain that the computer operates with RAM chips of all speeds. The method was limited to bank selection

since the CPU can only address 64K at any one given time. We chose to exchange the lower banks of 32K, address between zero and 32767 (0000-7FFF). Now with three lower banks of RAM, BASIC can have three programs resident in memory. Also there is another 32K bank of RAM, addresses 32768 to 65535 (8000-FFFF), where a control program can be placed to operate the lower three banks, similar to 128K operation of an Apple II. Realistically speaking, with ROM included, there is a total of 160K available. Refer to Figure 1 for a block diagram of how the 32K pages are configured.

Figure 1

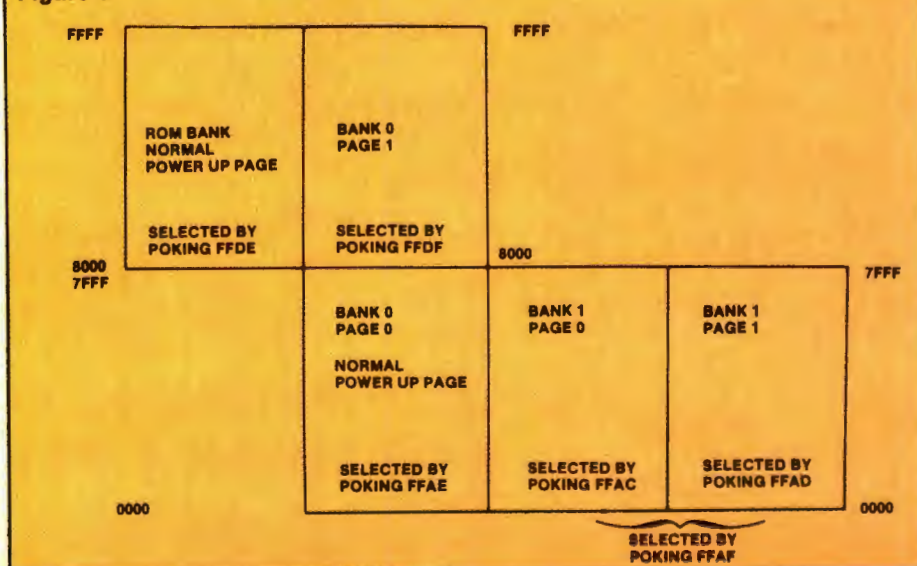
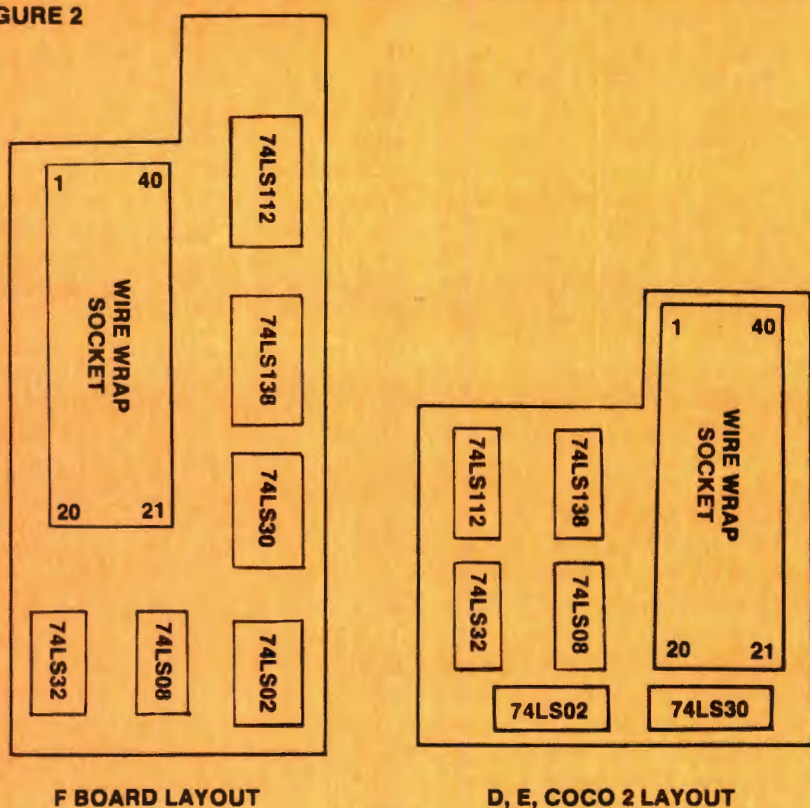


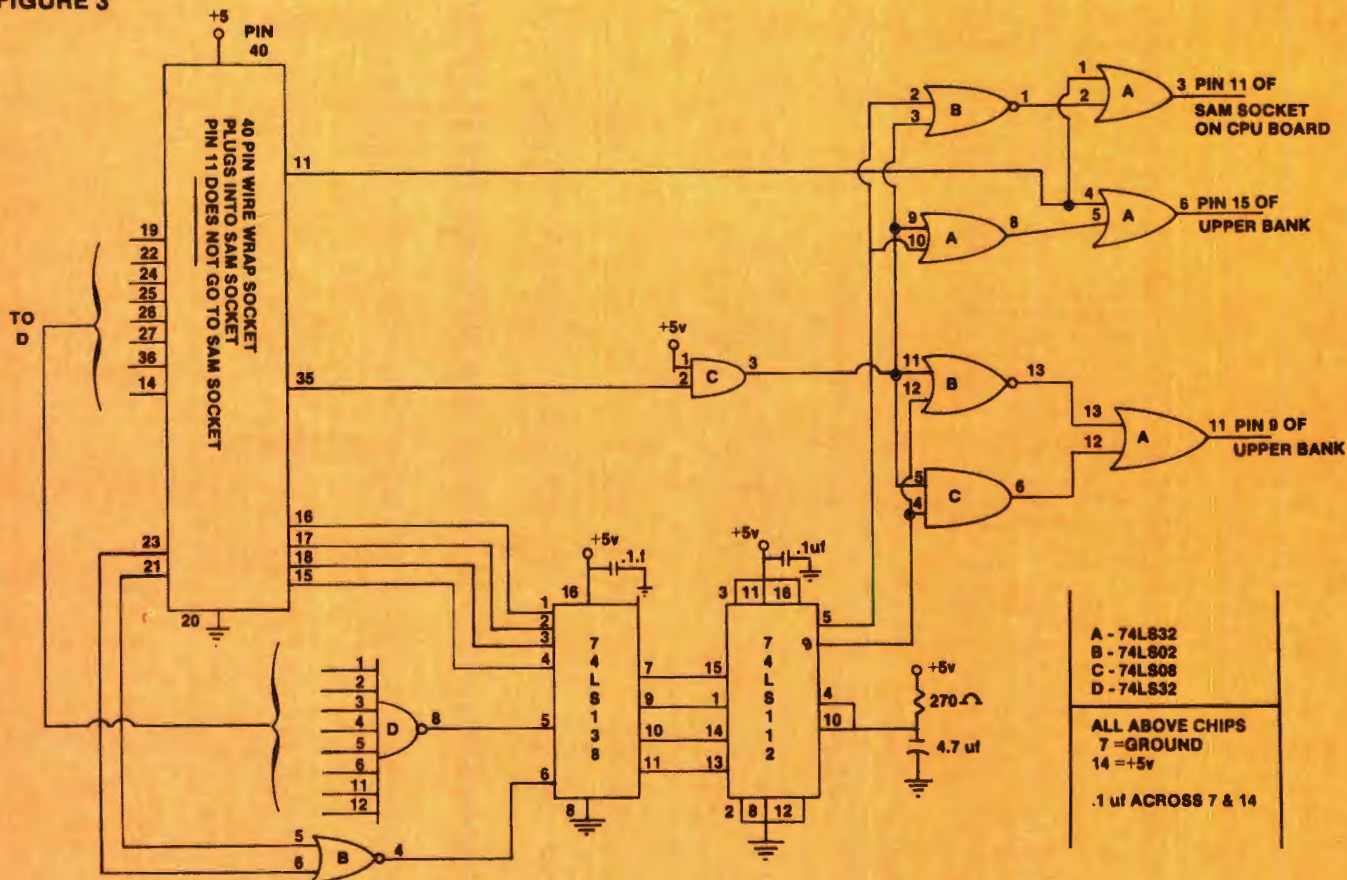
FIGURE 2



How To Do It

The simplest way of inserting memory, since there are only eight RAM sockets, is to piggyback the chips. There are data lines running through the 74LS244 (U19), that exist only at that chip. To place the additional RAM anywhere else would require the removal of the 74LS244 and installing a series of jumpers from its position. The drawback of this is since late version "E" boards, the 74LS244 has been soldered onto the board. Also the more wire that runs around inside the computer, the more noise (RF) the computer will generate. By piggybacking the 64K chips, these problems disappear. Most of the signals required to operate more memory will come from the SAM chip; to do this the SAM is elevated by means of a wire wrap socket. The additional chips are then placed on a board attached to that socket. Depending on your own level of soldering ability, there are a couple of ways to proceed. Using the suggested board layout (Figure 2), cut a sheet of perf board to dimension. Then follow the wiring diagram (Figure 3), and hand wire the board. The parts necessary to make 64K CoCo into a 128K CoCo are as follows:

FIGURE 3



8 - 64K RAM Chips (4164 or equiv.)
 1 - 74LS02 Quad NOR gate
 1 - 74LS08 Quad AND gate
 1 - 74LS30 8 Input gate
 1 - 74LS32 Quad OR gate
 1 - 74LS112 Dual JK Flip Flop
 1 - 74LS138 3 to 8 Line Decoder
 1 - 270 ohm Resistor
 1 - 4.7 uf Capacitor
 6 - .1 uf Capacitors
 1 - Spool of Wire (R/S # 278-501)

One important item is that pin 11 of the SAM chip does not go through to the original socket as all the other pins do. It should be cut off right below its connection to pin 4 of the 74LS32.

If you prefer the board with the six chips soldered and tested, it is available from DSL Computer Products for \$34.95. If you want to just plug in and go, the complete mod including an additional 64K of RAM is \$99. The installation of the complete mod is solderless.

How To Use It

Once the 128K modification is done, what can you do with it? When you power up the computer everything should be the same as normal including

the familiar answer to the ?MEM question. So how do you have 128K? Type in the command *POKE &HFFAE,0* and press ENTER. Now if you see garbage on the screen this is normal, press Reset, and the computer powers up again. If you don't have a disk system the computer will reset automatically, usually. You are now in bank zero, page zero of RAM. This is the normal bank in which a 64K Color Computer will power up. To enter the upper bank of 64K type *POKE&HFFAC,0:POKE&HFFAF,0*. Again, if you see garbage just press Reset. You are now in bank one, page zero. (One way to be sure you are switching banks is that the screen will change with the bank. If the poke has no effect you may have a wiring problem.) Finally, type *POKE&HFFAD,0*. This is bank one, page one. Now all three lower pages of 128K mod have been initialized.

Referring to Figure 1 again, the normal memory map has not been changed, but rather modified. Two 32K pages have been added along side the present 32K page used by BASIC. All current software will run as it normally does. The only way a different bank can be accessed is by poking (writing) to a memory location between 65452 and

65455. The value poked into that address can be zero to 255. By switching banks, three programs can reside in the computer at the same time. The CPU can only run one program at a time, unless another operating system is controlling it, such as FLEX or OS-9. However, all three programs can be run in such a way that it will seem that they are running all the time. There is one consideration using BASIC; that is the location of the Stack Pointer. If one of the programs clears space for strings or arrays, the Stack Pointer would be adjusted. Then when that bank is switched out the Stack Pointer would be pointing to nowhere. The results could cause the computer to lock up. For this reason there is a program listing included called *STAKSTAT*. This will initialize all three lower banks, transfer ROM to the upper bank of RAM, and add a new command to BASIC. By entering the command *PAGE x*, where x is 0, 1, or 2, that page will be selected and the correct stack value will be placed into the Stack Pointer. Please note that *STAKSTAT* can be used freely with any 128K software that you develop. However, if you wish to send it to a magazine to have it published, remember where you saw it first!

Listing 1:

```

00100 *****
00110 **
00120 ** PAGE VERSION 9.10.84 **
00130 **
00140 ** COPYRIGHT (c) 1984 **
00150 ** DSL COMPUTER PRODUCTS **
00160 ** WRITTEN BY **
00170 ** GERALD S. EGGART **
00180 **
00190 *****
00200
00210 *
00220 *THIS PROGRAM IS FOR USE WITH THE DSL 128K UPGRADE
00230 *UPON ENTRY (EXEC) ALL REGISTERS ARE MODIFIED
00240 *UPON CALL (PAGE) STACK IS PRESERVED & ALL REGISTERS MODIFIED
00250 *
7000 00260 ORG $7000 LOAD ADDRESS
00270
00280 *
00290 *INITIALIZE ZERO PAGE ROUTINE FOR NEW COMMAND
00300 *
00310
7000 CE #13E 00320 INITI LDW #013E THE ADDRESS WHERE THE INTERPRETER
00330 * VECTOR TABLE IS
00340 *
7003 04 01 00350 HRE LDA #1 ONE NEW COMMAND
7005 A7 C0 00360 STA ,U+
7007 30 0D 0024 00370 LEAX TABLE,PCR ADDRESS OF NEW COMMAND TABLE
7008 AF C1 00380 STX ,U++
700D 30 0D 0023 00390 LEAX SUBRO,PCR NEW COMMAND HANDLING ROUTINE ADRS
7011 AF C1 00400 STX ,U++
7013 6F C4 00410 CLR ,U NO SECONDARY FUNCTIONS ADDED
7015 0E D277 00420 LDW #0B277 SECONDARY FUNCTION HANDLING
00430 * ROUTINE (?SN ERROR)
7018 AF 43 00440 STX 3,U CLEAROUT NEXT TABLE ENTRY...
00450 * NO MORE COMMANDS ADDED
701A 6F 45 00460 CLR 5,U
701C AF 42 00470 STX 2,U
00480
701E 30 0D 0022 00490 LEAX ENTRY,PCR ADDRESS OF ROUTINE TO DO PAGING
7022 AF 0D 0001 00500 STX ADDR,PCR STORE IT IN THE JUMP TABLE
7026 39 00510 RTS
00520

00530 *
00540 * NEW COMMAND TABLE
00550 *
00560
7027 0000 00570 ADDR FDB #0
7029 0000 00580 ADD1 FDB #0
702B 0000 00590 ADD2 FDB #0
702D 0000 00600 ADD3 FDB #0
00610
00620 *
00630 * NEW WORD TABLE
00640 *
00650
702F 50 00660 TABLE FCC /PAG/
7032 C5 00670 FCB 'E+000
7033 00 00680 FCB #
00690
00700 *
00710 * NEW INTERPRETER ROUTINE FOR THE ADDED WORD(S)
00720 *
00730
7034 01 E4 00740 SUBRO CNPA #0E4 HIGHEST TOKEN VALUE USED
00750 * (#0E5 FOR DISK 1.1)
7036 25 03 00760 BLD LO IF ITS LOWER ITS A VALID TOKEN
7038 7E D277 00770 JMP #B277 NOT A VALID TOKEN SO ?SN ERROR
00780
703B 30 0D FFEB 00790 LO LEAX ADDR,PCR GET ADDR OF WERE TO GO TO
00800 * EXEC COMMAND
703F 00 E1 00810 SUBR1 SUBA #0E1 SET OFFSET BYTE TO WHICH COMMAND
7041 7E ADD4 00820 JMP #ADD4 GO LET ROM EXEC COMMAND (ENTRY)
00830
00840 *
00850 * THIS IS THE TRUE EXECUTION OF THE PAGE COMMAND
00860 *
00870
7044 00880 ENTRY
7044 0D 0700 00890 JSR #B700 EVALUATE AN EXPRESSION
00900 * >0 /<256 AND RETURN IT IN B
00910 CNPB 02 IS THE PAGE VALUE GREATER THAN 2
7049 22 29 00920 BHI ERROR IF IT IS THEN 'PC ERROR'
704B 50 00930 TSTB IS IT PAGE 0?
704C 26 07 00940 BNE CONTST NOPE THEN GOTO CONTSY
00950
00960 *
00970 * PAGE 0

```

```

00980 *
704E AE E4 00990 LDX ,S GET RTS ADDRESS OFF THE STACK
7050 7F FFAE 01000 CLR %FFAE FLIP TO PAGE ZERO
7053 20 16 01010 BRA ENDIT AND EXIT
01020
01030 *
01040 * PAGE 1 AND PAGE 2 TEST
01050 *
7055 C1 01 01060 CONST CMPB #1 IS IT PAGE ONE?
7057 26 0A 01070 BNE PG2 NO GOTO PG2
01080 *
01090 * PAGE 1
01100 *
01110
7059 AE E4 01120 LDX ,S GET RETURN ADDRESS
705B 7F FFAE 01130 CLR %FFAE GO TO PAGE 1
705E 7F FFAE 01140 CLR %FFAE
7061 20 00 01150 BRA ENDIT AND EXIT
01160 *
01170 * PAGE 2
01180 *
01190
7063 AE E4 01200 PG2 LDX ,S GET RETURN ADDRESS OFF STACK
7065 7F FFAE 01210 CLR %FFAE GOTO PAGE 2
7068 7F FFAE 01220 CLR %FFAE
01230
706D 100E 00F3 01240 ENDIT LDY >#F3 GET THE LOC OF THIS PAGES STACK
706F 1F 24 01250 TFR Y,S PUT STACK BACK
7071 AF E4 01260 STX ,S SAVE RETURN ADDRESS BACK ON STACK
7073 39 01270 RTS
01280 *
01290 * ERROR HANDLER IF PARAMETER IS BAD IN PAGE COMMAND
01300 *
01310
7074 01320 ERROR
7074 7E 8706 01330 JNP 00706 GOTO 'FC ERROR'
01340
01350 *
01360 * THE FOLLOWING CODE IS USED ONCE TO
01370 * SET UP THE COMPUTER UPON EXEC
01380 * THIS CODES MOVES ROM TO RAM
01390 * RELOCATES THE MAIN PROGRAM

```

```

01400 * AND INITIALIZES ALL THREE BANKS
01410 * TO A POWER UP STATUS
01420 *
01430
7077 B6 0134 01440 START LDA 0134
707A 01 00 01450 CNPA 000
707C 26 06 01460 BNE STCONT
707E CE 0134 01470 LDU 00134
7081 FF 7001 01480 STU INIT1+1
01490
7084 01 14 01500 STCONT CNPA 020
7086 26 0A 01510 BNE HERE
7088 06 E5 01520 LDA 00E5
708A 07 7035 01530 STA SUBRO+1
708D 04 E2 01540 LDA 00E2
708F 07 7040 01550 STA SUBR1+1
7092 1A 50 01560 HERE ORCC 0050 DISABLE INTERRUPTS
7094 CE 0000 01570 LDU 000000 START OF ROM
7097 0E FFDE 01580 LDX 00FFDE ROM ON LOCATION
709A 100E FFDf 01590 LDY 00FFDF ROM OFF LOCATION
01600 *
01610 * COPY ROM TO RAM
01620 *
709E EC C4 01630 LOOP LDD ,U GET TWO BYTES OF ROM
70A0 0F A4 01640 CLR ,Y GOTO ALL RAM MODE
70A2 ED C1 01650 STD ,U++ SAVE THE BYTES IN RAM
70A4 0F 04 01660 CLR ,X BACK TO ROM/RAM MODE
70A6 1183 FE00 01670 CNPU 00FE00 AT THE END OF ROM????
70AA 22 02 01680 BMI EXIT YEP... GOTO EXIT
70AC 20 F0 01690 BRA LOOP GO DO SOME MORE
01700
70AE CE F000 01710 EXIT LDU 00F000 DONE MOVING ROM TO RAM SO NOW
01720 * LETS MOVE THE PROGRAM ABOVE ROM
70B1 30 8D FF4B 01730 LEAX INIT1,PCR START OF PROGRAM
70B5 100E F000 01740 LDY 00F000 DESTINATION ADDRESS
70B9 7F FFDf 01750 CLR 00FFDF GOTO ALL RAM MODE
01760
70BC EC 01 01770 MLOOP LDD ,X++ GRAB 2 BYTES OF PROGRAM
70BE ED A1 01780 STD ,Y++ SEND IT ABOVE ROM
70C0 0C 7141 01790 CNPX 00DUMY END OF PROGRAM???
70C3 25 F7 01800 BLO MLOOP NO THEN GO GET SOME MORE
01810
01820 *
01830 * NOW GO TO THE ROUTINE INTPGE TO INITILAIIZE MEMORY.
01840 * FIGURE OFFSET
01850 *
01860
70C5 30 8D 000B 01870 LEAX INTPGE,PCR ADDRESS OF NEAR THE
01880 * ROUTINE NOW IS
70C9 1E 10 01890 TFR X,D
70CB 03 7000 01900 SUBD 01INIT1 SUBTRACT THE START OF THE PROGRAM
70CE C3 F000 01910 ADDD 00F000 ADD OFFSET
70D1 34 06 01920 PSHS D FAKE RETURN ADDRESS
70D3 39 01930 RTS GOTO INTPGE ROUTINE THAT HAS
01940 * BEEN TRANSFERED INTO UPPER MEMORY
01950 *
01960 *
01970 *
01980 * THIS ROUTINE INITIALIZES ALL LOWER 32K PAGES
01990 INTPGE LDY 00FFAC PAGE SELECT ADDRESS
70DB 06 F2 02000 LDA 00F2
70DA 07 00F3 02010 STA >#F3
70DD 7F FFAE 02020 CLR %FFAE
70DE 01 00F3 02030 CNPA >#F3
70E3 27 13 02040 BEQ CNTINT
70E5 CE 0000 02050 LDU 000
70E8 7F FFAF 02060 INITLP CLR %FFAF
70EB EC C4 02070 LDB ,U
70ED 7F FFAE 02080 CLR %FFAE
70F0 ED C1 02090 STD ,U++
70F2 1183 7FFF 02100 CNPU 007FFF
70F6 25 F0 02110 BLO INITCP
02120
70FB CE 0000 02130 CNTINT LDU 000 START COPY AT ZERO
70FB 0D 29 02140 BSR COPY 00 COPY A PAGE
70FD 100C FFAD 02150 CNPY 00FFAD ARE WE AT THE LAST 32K PAGE?
7101 27 04 02160 BEQ OUT YEP.... TIME TO GO!
7103 31 21 02170 LEAY 1,Y NO SO POINT Y TO NEXT MAGIC MEMORY LOCATION
7105 20 F1 02180 BRA CNTINT GO DO IT ALL AGAIN
02190
7107 7F FFAF 02200 OUT CLR %FFAF NOW TO COPY THE STACK ADDRESS
02210 * TO ALL PAGES
710A 7F FFAE 02220 CLR %FFAE
710D 1F 41 02230 TFR S,X GRAB THE STACK ADDRESS
710F 0F 00F3 02240 STX >#F3 SAVE IT IN PAGE 1
7112 7F FFAD 02250 CLR %FFAD
7115 0F 00F3 02260 STX >#F3 ALSO PAGE 2
7118 7F FFAE 02270 CLR %FFAE
711B 0F 00F3 02280 STX >#F3 AND PAGE 3
711E 1C AF 02290 ANDCC 00AF RENABLE INTERRUPTS

```

"Good Memory Database"

(FROM 9/84 RAINBOW REVIEW). CALL FOR FREE REPRINT!

DISK & PRINTER ARE OPTIONAL IN THE SUPER-FRIENDLY DO-FILE SYSTEM: NEEDS ONLY TAPE, 32K & EXT. BASIC. ALL WORK DONE IN MEMORY. NO PROGRAMMING REQUIRED TO BUILD HOUSEHOLD INVENTORY, WINE LIST, ANY LIST. YOU DESIGN YOUR OWN RECORDS. AND - YOU DON'T HAVE TO GET IT RIGHT THE FIRST TIME. EXPAND ANY FIELD. ADD NEW FIELDS - WITHOUT LOSING ANY EXISTING DATA. CREATE, REMOVE AND CHANGE RECORDS - SEARCH, SORT, LIST AND TOTAL THEM - LOAD AND SAVE FILES USING CASSETTE OR DISK (OR BOTH). DISPLAY YOUR RECORDS ON THE SCREEN AND/OR PRINTER IN ORIGINAL OR SORTED SEQUENCE (OR BOTH). SELECT AND SORT (ASCENDING OR DESCENDING) RECORDS ON UP TO 3 FIELDS. YOUR FILES CAN BE AS BIG AS 16,500 CHARACTERS. DO-FILE HAS EASY, ONE-KEY COMMANDS AND A HELP MENU, TOO. IT IS EVEN DESIGNED TO PROTECT YOU FROM MAKING MISTAKES! DO-FILE IS EDUCATIONAL - YOU LEARN FILE MANAGEMENT USING A 35-PAGE TUTORIAL GUIDE & SAMPLE DATA FILE!

CALL FREE! 1-800-334-0854, EXT. 856. TO ORDER. OR, SEND \$29.95 (CHECK OR MONEY ORDER) PLUS \$2.00 FOR POSTAGE/HANDLING (PA RESIDENTS ADD \$1.92 TAX), TO:

SOLID

SOFTWARE



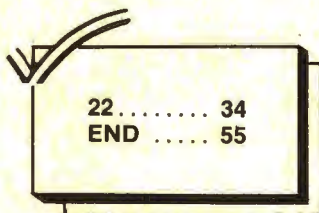
P.O. Box 712 • Levittown, PA 19058



```

7120 7F FFAE 02300 CLR 0FFAE
7123 7E F000 02310 JMP 0F000 GO INITIALIZE THE NEW WORD TABLE
                                02320 * IN PAGE 0 AND RETURN TO BASIC
                                02330 *
                                02340 * THIS ROUTINE COPIES ONE 32K PAGE TO ANOTHER
                                02350 *
7126 7F FFAE 02360 COPY CLR 0FFAE 8000 PAGE ZERO
7129 EC C4 02370 LDD ,U YANK A COUPLE OF BYTES
712B 7F FFAF 02380 CLR 0FFAF SELECT OTHER 64K BANK
712E 6F A4 02390 CLR ,Y SELECT WHICH 32K PAGE
7130 ED C1 02400 STD ,U++ SAVE THE BYTES
7132 1183 7FFF 02410 CMPL 007FFF END OF 32K PAGE???
7136 25 EE 02420 BLO COPY NO THEN DO SOME MORE!
7138 17 FEC5 02430 LBSR INIT1 YEP SO INITIALIZE THE NEW COMMAND
                                02440 * TABLE IN THIS PAGE
713B 7F FFAE 02450 CLR 0FFAE BACK TO PAGE ZERO FOR THE RTS
713E 39 02460 RTS
                                02470
                                02480 *
                                02490 * DUMP LABEL SO I KNOW WHERE THIS PROGRAM ENDS!!!
713F 0000 02500 STACK FDB 00
7141 0000 02510 DUMP FDB 00
                                02520 END START
00000 TOTAL ERRORS

```



Listing 2:

```

0 * *****
1 * **
2 * ** PAGER VERSION 9.10.84 **
3 * **

```

SuperSpooler

Free your fast computer from your slow printer!

SuperSpooler allows your computer to work on one task while your printer works on another.

- works with any size memory (64K recommended)
- compatible with Telewriter, Ultra 80CC etc., and all BASIC programs
- small - 64K version uses only 80 bytes of user RAM
- written in relocatable machine language
- 32K buffer with a 64K system
- selectable buffer size and location
- reprint buffer as many times as you want, when you want
- clear buffer at any time
- check status of buffer
- adjust speed of output to suit your printer and interface
- most features enabled with one keystroke at any time, even during application program execution

\$19.95 U.S. or **\$24.95 Canadian**

Also Available:
SBASIC 1.0—a structured BASIC pre-compiler.

\$19.95 U.S. or **\$24.95 Canadian**

Ordering Information:

Please specify cassette or disk. To order send check or money order, or use VISA or MasterCard. If using VISA or MasterCard, include card number and expiration date.

For UPS delivery add \$1 in Ontario and Montreal
\$3 in the U.S.

Ontario residents add 7% sales tax.

To order or for more information please write,

Tandar Software

12 Araman Drive Agincourt, Ontario Canada M1T 2P6 (416) 293-2014

DEALER INQUIRIES WANTED



```

4 * ** COPYRIGHT 1984 **
5 * ** DSL COMPUTER PRODUCTS **
6 * ** **
7 * *****
8 CLEAR200, &H7000
9 FORI = 28672 TO 28976
10 READ A
11 POKE I, A
12 NEXT I
13 EXEC 28767
14 CLEAR200, &H7FFF
15 NEW
16 DATA 191,224,0,16,191,224,2,2
55,224,4,253,224,6,189,183,11,19
3,2,34,59
17 DATA 31,65,191,0,243,93,38,7,
174,228,127,255,174,32,22,193,1,
38,10,174
18 DATA 228,127,255,172,127,255,
175,32,8,174,228,127,255,173,127
,255,175,16,190,0
19 DATA 243,31,36,175,228,190,22
4,0,16,190,224,2,254,224,4,252,2
24,6,57,190
20 DATA 224,0,16,190,224,2,254,2
24,4,252,224,6,126,183,6,23,0,36
,127,255
21 DATA 223,48,141,0,23,16,142,1
70,241,141,8,142,240,0,191,171,1
73,32,44,166
22 DATA 128,39,4,167,160,32,248,
57,80,65,71,69,160,0,26,80,206,1
28,0,142
23 DATA 255,222,16,142,255,223,2
36,196,111,164,237,193,111,132,1
7,131,254,0,34,2
24 DATA 32,240,57,206,240,0,48,1
41,255,86,16,142,240,0,127,255,2
23,236,129,237
25 DATA 161,140,113,47,37,247,48
,141,0,11,31,16,131,112,0,195,24
0,0,52,6
26 DATA 57,16,142,255,172,134,24
2,183,0,243,127,255,174,177,0,24
3,39,19,206,0
27 DATA 0,127,255,175,236,196,12
7,255,174,237,193,17,131,127,255
,37,240,206,0,0
28 DATA 141,39,16,140,255,173,39
,4,49,33,32,241,127,255,175,127,
255,172,31,65
29 DATA 191,0,243,127,255,173,19
1,0,243,127,255,174,191,0,243,28
,175,127,255,174
30 DATA 57,127,255,174,236,196,1
27,255,175,111,164,237,193,17,13
1,127,255,37,238,127
31 DATA 255,174,57,0,0

```